CASCADE CONTROL OF UNSTABLE SYSTEMS WITH APPLICATION TO STABILIZATION OF SLUG FLOW

Espen Storkaas * **Sigurd Skogestad** *,1

* *Department of Chemical Engineering, Norwegian University of Science and Technology, Trondheim, Norway*

Abstract: The topic of this paper is the effect of stabilizing control on the remaining control problem. In many cases there is no effect. However, stabilization requires the active use of inputs, and the underlying unstable pole will appear as an undesirable unstable zero if we are concerned with input performance. The implications of this are clearly demonstrated on the application to stabilization of severe slugging in two-phase pipeline-riser systems. We find that a controllability analysis gives important information for measurement selection and performance limitations.

Keywords: Cascade control, unstable, instability, controllability, performance limitations

1. INTRODUCTION

When unstable (RHP) poles are present in a system, these need to be stabilized, preferably by low-level (secondary) control. Skogestad *et al.* (2002) discuss the control limitations imposed by a RHP pole with focus on the stabilizing control loop. They show that a RHP pole imposes a lower limit on the $\mathcal{H}_2$ - and $\mathcal{H}_\infty$- norm of the transfer function KS from outputs to inputs, and that the unstable pole manifests itself as a RHP-zero in KS, limiting input movement.

The stabilizing loop needed when a RHP pole is present is usually part of a larger control system. The topic of this paper is the effect unstable poles and its manifestations have on the higher levels in the system. We will briefly discuss the open loop behavior for a process with stabilized RHP poles, and study in more detail a cascade control system where the inner loop stabilizes the unstable poles of the system.

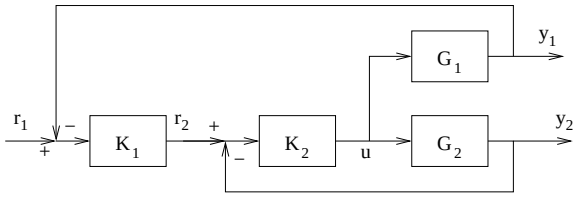


Fig. 1. Block diagram for a cascade control system, 1 - primary (outer) loop, 2 - secondary (inner) loop

We will use the cascade configuration given in Figure 1. $G_1(s)$ and $G_2(s)$ are the transfer functions from the input $u(s)$ to the primary output $y_1(s)$ and the secondary output $y_2(s)$, respectively. We will omit the argument (s) in the following to simplify the notation. We assume that G_2 has at least one unstable pole. K_2 is the controller for the inner loop, and K_1 is the controller for the outer loop. The output y_1 from K_1 is the reference signal to K_2. Both controllers are assumed stable.

[1] Author to whom correspondence should be addressed. email: skoge@chemeng.ntnu.no

Two different cases will be studied, one where the unstable modes are detectable (G_1 contains the same unstable poles as G_2), and one where the unstable modes are not detectable in y_1 (G_1 is open-loop stable). Our main example will be the stabilization of gravity induced slug flow in multiphase pipeline-riser systems. This system is excellent for demonstrating the importance for controllability analysis and the limitations imposed by non-minimum-phase systems, as it contains both RHP poles and RHP zeros as well as other control limitations depending on the choice of controlled outputs. It also contains both the alternatives for G_1 described above.

2. LIMITATIONS IMPOSED BY UNSTABLE POLES AND ZEROS

Consider a plant with state space matrices A, B, C and D, and transfer function $G(s) = C(sI - A)^{-1}B + D$. The poles of the plant are the eigenvalues of A, and the plant is unstable if the poles are in the RHP plane. By the right-half plane we mean the closed right half of the complex plane, including the imaginary axis (jω-axis).

Unstable poles need feedback for stabilization, the presence of RHP poles places a lower band on the bandwidth of the feedback system. For a real pole p, Skogestad and Postlethwaite (1996) gives the lower limit $\omega_c > 2p$, while for an imaginary pole the limit is $\omega_c > 1.15|p|$.

Zeros usually arise when competing effect, internal to the system, results in a zero output from a transfer function for non-zero inputs. We will later show that RHP zeros also can arise from stabilization of RHP poles. For a SISO system, the zeros z_i are the solutions to $G(z_i) = 0$. RHP zeros give rise to inverse response behavior, as the output from a stable plant with n_z RHP zeros will cross its original value n_z times as response to a step change in its input (Holt and Moriari, 1985)(Rosenbrock, 1970).

It is also well known form classical root-locus analysis that as the feedback gain increases towards infinity, the closed loop poles moves towards the open-loop zeros. This implies high gain instability and bandwidth limitations. Skogestad and Postlethwaite (1996) derives the following upper bandwidth limitations for systems with RHP zeros.

Real zero:

$$\omega_B \approx \omega_c < \frac{z}{2} \tag{1}$$

Complex zeros

$$\omega_B \approx \omega_c < \begin{cases} |z|/4 & Re(z) \gg Im(z) \\ |z|/2.8 & Re(z) = Im(z) \\ |z| & Re(z) \ll Im(z) \end{cases} \tag{2}$$

When both RHP poles and zeros are present in a system, the presence of the above mentioned upper and lower bandwidth limitations may render stabilization impossible. To see this consider the effect of a pair of unstable complex poles with dominant imaginary part ($Re(p) \ll Im(p)$) and magnitude $|p|$ and a single real RHP zero z. For the bandwidth limitations to be met, we must approximately require

$$z > 2.3|p| \tag{3}$$

in order to get acceptable performance and robustness.

3. FUNDAMENTAL ALGEBRAIC LIMITATIONS

With the inner stabilizing loop (K_2) closed, the transfer function for the remaining control problem becomes:

$$G = G_1 S_2 K_2 \tag{4}$$

$$S_2 = (I + K_2 G_2)^{-1} \tag{5}$$

where G is the transfer function from r_2 to y_1. To ensure internal stability, unstable poles in G_2 cannot be cancelled by K_2. Then, by (5), S_2 must have RHP-zeros in the same location as the unstable poles in G_2 if internal stability is to be achieved. If the unstable modes are observable in y_1 (G_1 contains the same unstable poles as G_2), the RHP-zeros in S_2 will be cancelled in G. In this case, any bandwidth limitations due to RHP-zeros must come from G_1.

If the unstable modes in G_2 are not present in G_1, as with input reset in the outer loop $y_1 = u$ ($G_1 = I$), G will have RHP zeros at the location of the RHP poles of G_2. These RHP zeros will limit the bandwidth of the higher level control system, i.e. for the outer loop in the cascade system in Figure 1.

4. IMPLICATIONS OF STABILIZED RHP POLES

If the RHP-zeros originating from the unstable poles of G_2 are not cancelled by G_1, G will have RHP zeros resulting in an inverse response through the process. The paradox for control is that the slower the instabilities in G_2 (easy stabilization), the slower the inverse response through

G and the lower the allowed bandwidth (slow control) in the higher levels in the control hierarchy. In other words, the harder job the inner loop controller has, the better control can be expected from the outer loop.

Consider a process G_2 with a real unstable pole located at p. The sensitivity function S_2 for the stabilized system will have a RHP zero at p. The bandwidth limitations for the outer loop in the cascade will in our case be $\omega_B \approx p/2$ (see (1)). This confirms that for slow instabilities (p small) the bandwidth limitations are more severe than for fast instabilities (p large).

A practical example can be found by trying to keep your balance on a bicycle while staying at the same place. This is obviously an unstable system, and you will have to use your body as a controller to hold your balance. You will find it easier to keep your balance the more the bike is tilted over to the side, as the instability gets slower. However, since you use your body to stabilize the unstable bike, you will find that there is an inverse response in trying to tilt the bike.

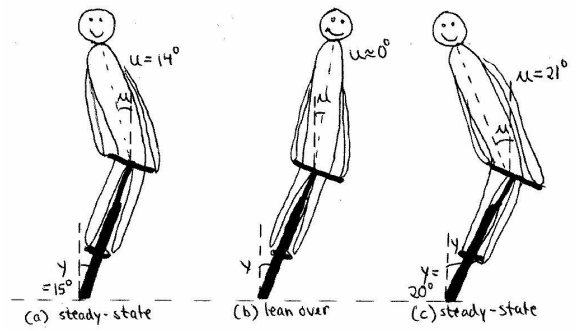


Fig. 2. Inverse response for a bicycle caused by an underlying instability

Consider Figure 2 where the aim is to tilt the bike from an initial angle $y = 15°$ (Fig. 2a) using your body (u) to an angle $y = 20°$ (Fig. 2c). Because of the inverse response, you first have to tilt your body in the direction of the tilt to start the movement (Fig. 2b). Eventually, you will have to move your body back to restore balance. This inverse response will be slower the greater the angle y, changing the angle while keeping balanced gets progressively slower as the tilting angle is increased.

5. CASE STUDY

Multiphase flow in pipelines differs from regular single phase flow in that a wide variety of flow patterns, also called flow regimes, can develop, dependent on the flow rates, fluid properties and pipeline geometry. Gravity-induced slug flow occurs as a result of a lowpoint connected to an inclining section of the pipe. The pressure drop in the pipeline and the interphase friction between the phases are in these cases not sufficient to transport the liquid uphill in a stationary fashion. The liquid will accumulate in the lowpoint, and a liquid slug will form.

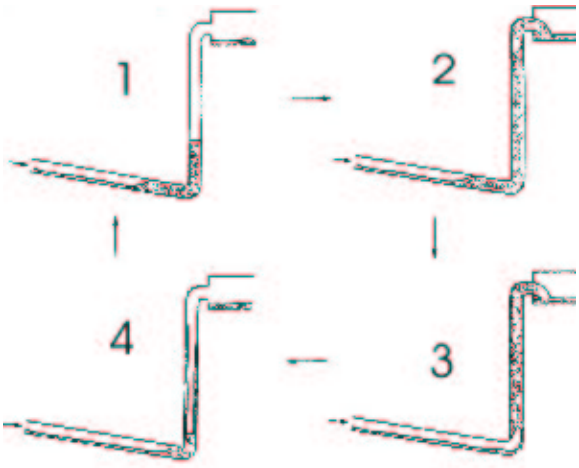


Fig. 3. Illustration of the cyclic behavior (slug flow) in pipeline-riser systems

The liquid slug that forms will block the flow of gas in the pipe, and grow until enough upstream pressure has developed to overcome the weight of the liquid slug. An illustration of the slug cycle is given in Figure 3. In pipeline-risers systems in the offshore oil industry, these slugs can grow very large, and cause severe problems when they are delivered to the downstream production facility. The inlet separator on the platform will experience large level variation, resulting in poor separation and in some cases flooding. Load variations on the compressors may lead to unnecessary flaring. Another aspect is that the pressure variation caused by slug flow might lead to reduced well performance.

Stabilizing the flow using active control has a great economic potential both in improved regularity and in the possibility for increased recovery of oil. For earlier work on stabilizing slug flow, please consult Hedne and Linga (1990), Henriot *et al.* (1999) and Havre *et al.* (2000).

5.1 Dynamic model of gravity induced slug flow

We have developed a simplified nonlinear model with 3 states that describes the process (Storkaas *et al.*, 2003). The simplified model is more suitable for control analysis than the conventional PDE-based models used to describe these systems, as it is continuous, low-dimensional, and relatively easy to tune. Storkaas *et al.* (2003) shows that the model approximates the physical behavior of these systems with sufficient accuracy to be used for controller design and analysis.

The only actuator for theses systems are usually the downstream production choke. Possible measurement alternatives are upstream pressure (P_1), downstream pressure (P_2) and density (ρ_T) measured just upstream the actuator and volumetric (Q) and mass flow (W) through the production choke (u = Valve opening). The major disturbances is the feed flow, the feed liquid fraction and the downstream separator pressure. We have added first order dynamics to the actuator.

5.2 Stability analysis

The bifurcation diagram for the process is given in Figure 4, with valve opening on the horizontal axis and upstream pressure P_1 at the vertical axis. The solid lines represent open loop stable operation, while the dashed lines indicate unstable operation. Two solid lines for a given valve opening represent a limit cycle were the maximum and minimum pressures are given.

As seen from Figure 4, the process is stable when operated with low choke openings. When the choke opening is increased above $u = 0.13$, the process goes through a Hopf bifurcation, resulting in an open loop stable limit cycle (slug flow). However, as seen from the dashed line in Figure 4 there exists unstable, stationary operating points (with a pair of complex RHP poles) for valve openings greater than $u = 0.13$. The control problem is thus to design a control system that stabilizes this mode of operation.

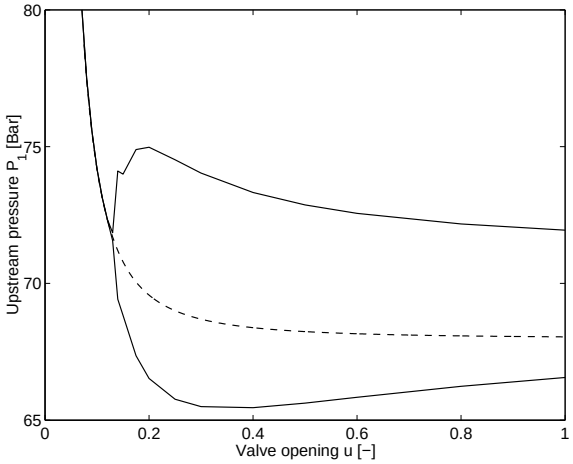


Fig. 4. Bifurcation diagram for the pipeline-riser system

5.3 Controllability analysis

The following analysis is performed on a linearized model obtained around a typical unstable operating point (along the dashed line in Figure 4). The same analysis performed at other operating point show that the results obtained below are typical for this system. It should also be noted that only operating points with valve openings in the low-to-medium range is relevant for closed loop operation. The reason for this is that the system typically is designed with a valve that does not restrict the flow when it is fully opened. A consequence of this is that the pressure drop over the valve at high openings is low, resulting in insufficient process gain for stabilization in this region.

P_1	P_2	ρ_T	Q	W
-0.0034	0.0142	-0.0004	-4.1173	-7.6323
-	3.2489	0.0048	-0.0042	-0.0004
-	-	-	-0.0004	0

Table 1. Zeros for different measurement alternatives. Positive (RHP) zeros imply control problems

5.3.1. Measurement selection for stabilizing control (y_2) We consider an unstable operating point corresponding to a valve opening of u=0.175. The poles of the system at this operating point is -6.11 and $0.0008 \pm 0.007i$. The zeros for the different measurement alternatives in Section 5.1 are given in Table 1. The locations of the different measurement alternatives are illustrated in Figure 5.

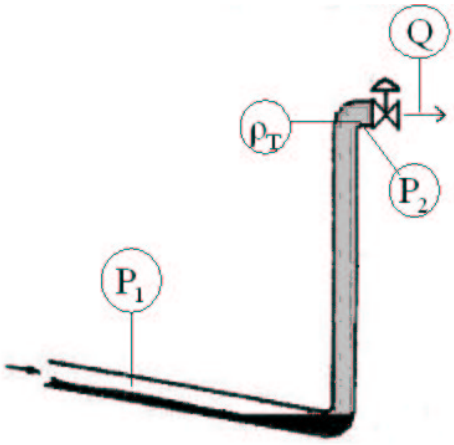


Fig. 5. Measurement locations in pipeline-riser systems

The upstream pressure, P_1, contains a single left half plane (LHP) zero. This imposes no fundamental control limitations, so P_1 would thus be a good choice for y_2. However, this measurement can in many cases be either unreliable or unavailable, and other measurements should also be considered.

Alternatives P_2 and ρ_T in Table 1 both have RHP-zeros. With the bandwidth limitation given above (see (3)), we know that we cannot have RHP zeros

smaller than approximately 0.016 for acceptable performance. Thus, neither P_2 nor ρ_T are suitable as measurements for a stabilizing control due to the bandwidth limitations imposed.

Volumetric flow Q or mass flow W are better alternatives, but both have LHP zeros close to or at the imaginary axis. This means that the steady state gain from u to Q or W is close to or identically zero. Physically, the outflow must equal the inflow at steady state, and the outflow (Q or W) cannot be set independently. By closing the loop from u to Q or W (i.e. with a P-controller), we may be able to stabilize the system, but we cannot affect its steady-state behavior, and the system will "drift". This drift may be avoided by measuring another (primary) variable and using a cascade configuration as discussed next.

We choose to control the volumetric flow Q ($y_2 = Q$) in the inner loop. We close this loop with an PI controller K_2 with gain $+8\ bar^{-1}$ and integral time 40 s (chosen to match the time constant of the valve). The integral action is added under the assumtion that a cascade system is used. With reference to Figure 1 and (4), (5), this results in the following transfer functions:

$$G_2 = \frac{0.00247\,(s+4.117)\,(s+0.0042)\,(s+0.0004)}{(s+0.025)\,(s+6.112)\,(s^2-0.0016s+0.00005)} \quad (6)$$

$$S_2 = \frac{s\left(s^2-0.0016s+0.00005\right)}{(s+0.0002)\,(s^2+0.011s+0.0001)} \quad (7)$$

$$G = G_1 S_2 K_2 = G_1 \frac{8\,(s+0.025)\left(s^2-0.0016s+0.00005\right)}{(s+0.0002)\,(s^2+0.011s+0.0001)} \quad (8)$$

5.3.2. Choice of primary control variable y_1 As mentioned, we here assume that we have closed the inner flow control loop ($y_2 = Q$). The above transfer functions show that the complex RHP poles in G_2 manifest themselves as RHP zeros in S_2. The choice of measurement in the outer loop (y_1) will determine if the RHP-zeros will appear in $G = G_1 S_2 K_2$ and thus be a problem for control in the outer loop. Disregarding the choice $y_1 = P_1$ (not measured), we see from Table 1 that $y_1 = P_2$ is probably the best alternative. For this choice

$$G_1 = \frac{-0.00007\,(s-3.249)\,(s-0.0142)}{(s+0.025)\,(s+6.112)\,(s^2-0.002s+0.00005)} \quad (9)$$

and the unstable poles in G_1 will cancel the unstable zeros in S_2, and G will not contain any RHP zeros due to the unstable poles in G_2. However, the RHP zero at $z = 0.0136$ in G_1 itself remains. With the bandwidth limitations caused by a real RHP zero given in (1), the closed loop cascade system has an approximate bandwidth limitation of $\omega_B \approx 0.0068$ for the outer loop.

One could also choose to use the choke valve position ($y_1 = u$) as a measurement. In this case, $G_1 = 1$, and the RHP zeros in S_2 appear in G. Now the allowed bandwidth in the outer loop will depend on the frequency of the instability. For the current operating point, the unstable poles of G_2 is $p = 0.0008 \pm 0.007i$. From (2), the approximate bandwidth limitation for this operating point is $\omega_B \approx 0.007$. A change in operating point would result in a change in the bandwidth limitations. For example, for the operating point with a set point in the outer loop of $u = 0.25$, G_2 has RHP poles on $0.0028 \pm 0.009i$ resulting in a bandwidth limitation $\omega_B \approx 0.0094$.

Comparing $y_1 = P_2$ and $y_1 = u$ we really have a choice among "two evils"', as the bandwidth will be limited in either cases.

5.4 Simulation of cascade control of pipeline-riser system

The above controllability analysis is confirmed by nonlinear simulations on the simplified model. We use a cascade control system with $y_2 = Q$ in the inner stabilizing loop and $y_1 = P_2$ or $y_1 = u$ in the outer loop.

The simulations in Figure 6 and 7 are started up in open loop, as can be seen from the initial oscillatory behavior. The controllers are turned on at $t = 0.5$ hrs, with a set point corresponding to the operating point with $u = 0.175$ as stationary value. The dashed lines represent the set points. At $t = 2.5$ hrs, the set point is changed to that corresponding to a valve opening of $u = 0.25$.

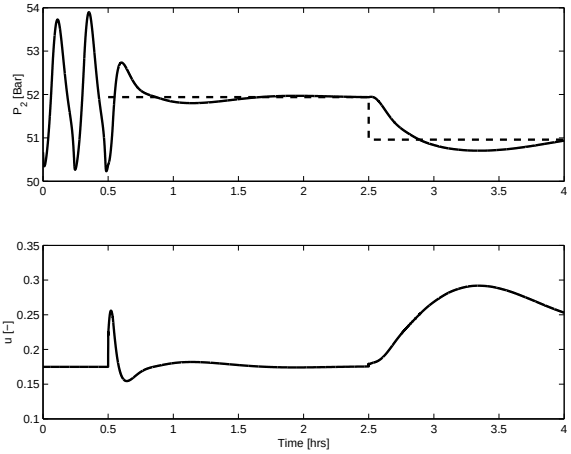


Fig. 6. Stabilizing control with $y_1 = P_2$, $y_2 = Q$

$y_1 = P_2$ is the controlled output in Fig. 6. The controller manages to stabilize the process at both operating points, but the response is rather sluggish, especially for the last operating point. This is due to the bandwidth limitations imposed by the RHP zero in G_1.

In Fig. 7, the valve position is the controlled output ($y_1 = u$). The speed of the response for

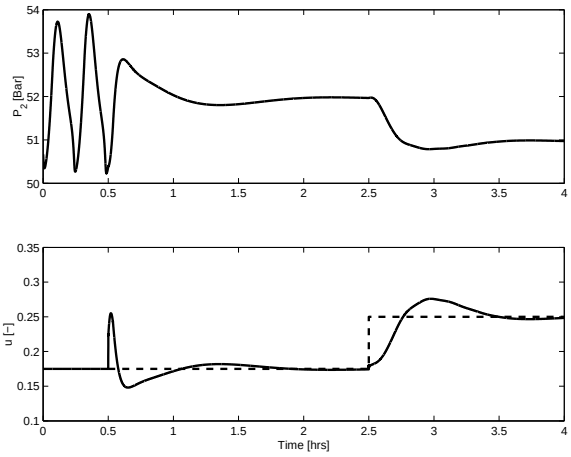


Fig. 7. Stabilizing control with $y_1 = u$, $y_2 = Q$

the first operating point is about the same as for the other controller, where we showed above that the bandwidth limitations were about the same. The response is faster for the second operating point, where the RHP poles in G_2 are faster.

5.5 Discussion

The bandwidth limitations for the outer loop in cascade control structure used above depends on the choice of y_1. For $y_1 = P_2$, the limitations arise due to the process transfer function in the outer loop, while for $y_1 = u$, the reason is a RHP zero from the stabilized unstable pole in G_2. For the studied operating point, the upper bandwidth limit for the two cases are similar. The location of the RHP zeros will be dependent on the operating point, but the dependency is much stronger in the case with $y_1 = u$. The unstable pole in G_1 that gives rise to the bandwidth limitation in this case is a strong function of operating point, whereas the RHP zero in G_1 with $y_1 = P_2$ is almost constant. The controllers must be tuned to deal with the worst-case bandwidth limitations, but the dependency of the RHP zeros to the operating point is nevertheless interesting.

In tuning the controllers, our aim has been to minimize the peaks on the closed loop sensitivity functions S and T. The reason for choosing this design target is that we want to maximize robustness. We assume an initial state on the limit cycle for the process, which might be far away from the desired operating point. Hence, it is important to be able to bring the process from a wide range of initial conditions into the unstable operating point. For that reason we try to maximize robustness. We have achieved values for M_S and M_T in the range 1.3-1.6 for the outer loop.

It should also be mentioned that none of the above mentioned RHP zeros are pinned to a certain output, so a multivariable (1 input, multiple outputs) controller (i.e. LQG) would not experience the same bandwidth limitations as the cascade control system discussed above.

6. CONCLUSIONS

We have shown that the sensitivity function through a stabilized system will have RHP zeros resulting in inverse response. The RHP zeros may also results in bandwidth limitations for higher level control loops, unless the RHP zeros are cancelled by the same instability in the process through the higher level loops. Slow RHP poles close to the imaginary axis are easy to stabilize but the resulting RHP zeros will be just as slow. In some cases it may lead to improved performance for the control system if one chooses to operate in a more unstable operating point.

The application to stabilization of severe slugging shows that the controllability analysis gives important information for measurement selection and performance limitations.

REFERENCES

Havre, K., K.O. Stornes and H. Stray (2000). Taming slug flow in pipelines. *ABB review* **4**, 55–63.

Hedne, P. and H. Linga (1990). Supression of terrein slugging with automatic and manual riser choking. *Advances in Gas-Liquid Flows* pp. 453–469.

Henriot, V., A. Courbot, E. Heintze and L. Moyeux (1999). Simulation of process to control severe slugging: Application to the dunbar pipeline. SPE Annual Conferance and Exibiton in Huston,Texas. SPE 56461.

Holt, B.R. and M. Moriari (1985). Design of resilient processing plants vi - the effect of right plane zeros on dynamic resilience. *Chemical Engineering Science* **40**, 59–75.

Rosenbrock, H.H. (1970). *State Space and Multivariable Theory*. Nelson, London.

Skogestad, S. and I. Postlethwaite (1996). *Multivariable feedback control*. John Wiley & sons.

Skogestad, S., K. Havre and T. Larsson (2002). Control limitations for unstable plants. In: *Proc. of IFAC World Congress, Barcelona, 21-26 July 2002, Paper T-Fr-M-15-1.*

Storkaas, E., S. Skogestad and J.M. Godhavn (2003). A low-dimentional model of severe slugging for controller design and analysis. In: *Proc. of MultiPhase '03, San Remo, Italy, 11-13 June 2003.*

COMBINED REAL-TIME AND ITERATIVE LEARNING CONTROL TECHNIQUE WITH DECOUPLED DISTURBANCE REJECTION FOR BATCH PROCESSES

In Sik Chin [,1] **S. Joe Qin** [,2] **Kwang S. Lee** [,3]
Moonki Cho

Department of Chemical Engineering, University of Texas at Austin, Austin, TX 78712, USA
Department of Chemical Engineering, Sogang University, Sinsoo-dong 1, Mapo-gu, Seoul 121-742, Korea

Abstract: A novel stochastic control framework for batch and repetitive processes is proposed. The framework provides a pertinent means to incorporate real-time feedback control (RFC) into iterative learning control (ILC) so that the performance of ILC is virtually decoupled from that of RFC. This is a new advancement since the currently practiced methods for combined RFC and ILC have su ered from the problem that RFC has undesirable e ects on ILC such as digression from its convergent track along the run index when there occur run-independent real-time disturbances. Performance of the proposed technique has been demonstrated in two numerical processes.

Keywords: Run-to-run Control, Iterative Learning Control, Model Predictive Control, Stochastic Control.

1. INTRODUCTION

Iterative learning control (ILC) is a relatively new technique that has been developed to improve the tracking performance of a process that executes the same operation repeatedly. For the past two decades since the Arimoto's contribution (Arimoto *et al.*, 1984), ILC methods have been steadily improved from SISO (single-input single-output) modeless deterministic approach to MIMO (multi-input multi-output) model-based stochastic approach and the application areas have been extended from the mechatronic systems like robots and disk drivers (Arimoto *et al.*, 1984; Bien and Huh, 1989) to chemical processes (Lee *et al.*, 1994), microelectronic systems (Lee *et al.*, 2001; Qin *et al.*, 2002), biomedical problems (Good *et al.*, 2002), and so forth. It is anticipated that the application of ILC will be broadened more because of the growing importance of batch or run-to-run operation in various industrial processes by the recent trend to produce small quantity of high-valued multiple products.

Basically, ILC concerns the issue of learning from the past operations in order to attain the ultimate tracking performance under model uncertainty and run-wise repeated disturbances. In real applications, however, it is desirable to treat the disturbance within a run and thus real-time feed-

[1] This work was supported by the Pose-doctoral Fellowship Program of Korea Science & Engineering Foundation (KOSEF).
[2] Corresponding author: phone (1-512)471-4417, fax (1-512)471-7060, e-mail qin@che.utexas.edu
[3] Or: phone (82-2)705-8477, fax (82-2)3272-0319, e-mail kslee@ccs.sogang.ac.kr

back control (RFC) need to be combined with ILC.

There are di erent methods to combine RFC with ILC. A widely practiced method is to add a feedback control block to ILC such that

$$u_k(t) = u_{k\ 1}(t) + H_1(t)e_{k\ 1}(1:N) + H_2(t)e_k(1:t) \quad (1)$$

where $(i:j)$ means data from $t=i$ to j; k represents the run index; H_1 and H_2 represent the gains for ILC and FBC, respectively. In fact, RFC-ILC can be realized without H_1 since $u_{k\ 1}(t)$ already contains the information of $e_{k\ 1}(t)$. Nevertheless, (1) represents the most general form of RFC-ILC.

A trouble with existing combined RFC-ILC techniques is that they lack the capability of distinguishing the run-independent real-time disturbance from the run-wise persisting disturbance. Both RFC and ILC try to reject the disturbances, in real-time for RFC and run-wise for ILC, respectively. When there occurs a large run-independent disturbance in the k^{th} batch, $u_k(t)$ may change excessively to reject the disturbance, and deteriorate the control performance of the following runs since $u_k(t)$ acts as a feedforward input signal for the next run. To the authors' knowledge, it seems that there has been no attempt to decouple ILC from RFC such that ILC deals with only the disturbance with strong run-wise correlation while RFC handles the run-independent disturbance.

The purpose of the research is to develop a novel framework for combined ILC and RFC, that can separately handle the run-independent real-time disturbance and the run-wise correlated disturbance. For this, a quite general form of disturbance model is rst assumed in a stochastic framework by decomposing the disturbance into three parts: the run-wise persisting part, the run-independent part, and the measurement noise. Each of them is separately handled by either ILC or RBC with the aid of the Kalman lters. To complete this, not only the state but also the input is split into two parts: one for ILC and the other for RFC. Through this approach, the resulting controller is able to appropriately discriminate the real-time disturbance from the run-wise persisting disturbance and prevents the e ects of the real-time disturbance from being carried over to the future runs. To realize the above concept, we propose a two-stage technique where RFC and ILC are executed in turn during and after a batch run. As a prototypical algorithm, we revised QILC (Quadratic criterion-based ILC) (Lee *et al.*, 2000) and BMPC (Batch Model-based Predictive Control) (Lee and Lee, 1997) and combine them into a two-stage algorithm.

2. DISTURBANCE PROPAGATION IN EXISTING TECHNIQUES

2.1 Process Modelling

We consider a linear discrete-time batch process with $u(t)$ and $y(t)$ as input and output variables, respectively, de ned over a nite interval with N sampling steps. Although the process undergoes dynamics within a batch run, it can be represented as a linear algebraic system which relates the input sequence vector to the output sequence vector over the underlying discrete-time domain.

$$\mathbf{y} = \mathbf{G}\mathbf{u} \quad \mathbf{d} \quad (2)$$

where

$$\mathbf{y} = \left[y^T(1)\ y^T(2)\cdots y^T(N)\right]^T \quad (3)$$

and likewise for $\mathbf{u}$ and $\mathbf{d}$. In the above, $d(t)$ represents the e ects of all possible uncertainties including the disturbance, model error, and bias term. For stochastic ILC design, it is sensible to decompose $\mathbf{d}$ into two terms: a run-wise correlated part and a run-independent part.

$$\mathbf{d} = \mathbf{w} + \mathbf{v}. \quad (4)$$

If we represent the run-wise correlated part as an integrated white noise process along the run index, then $\mathbf{d}_k$ can be expressed as follows:

$$\begin{aligned} \mathbf{w}_k &= \mathbf{w}_{k\ 1} + \Delta\mathbf{w}_k \\ \mathbf{d}_k &= \mathbf{w}_k + \mathbf{v}_k \end{aligned} \quad (5)$$

where both $\{\Delta\mathbf{w}_k\}$ and $\{\mathbf{v}_k\}$ represent the zero-mean white noise processes along k with covariance matrices $\mathbf{R}_{\Delta w}$ and $\mathbf{R}_v$, respectively.

Let $\mathbf{e}_k = \mathbf{y}_d \quad \mathbf{e}_k$ and $\bar{\mathbf{e}}_k = \mathbf{e}_k \quad \mathbf{v}_k$ where $\mathbf{y}_d$ is the desired output trajectory. Then the following inter-run transition model of tracking error can be derived from (2) and (5):

$$\begin{aligned} \bar{\mathbf{e}}_k &= \bar{\mathbf{e}}_{k\ 1} \quad \mathbf{G}\Delta\mathbf{u}_k + \Delta\mathbf{w}_k \\ \mathbf{e}_k &= \bar{\mathbf{e}}_k + \mathbf{v}_k. \end{aligned} \quad (6)$$

2.2 Pure ILC

The pure ILC algorithm can be written as

$$\Delta\mathbf{u}_k = \mathbf{H}_1\bar{\mathbf{e}}_{k\ 1} \quad (7)$$

In practice, $\bar{\mathbf{e}}_{k\ 1}$ is not measured, hence should be replaced by an estimate. (7) represents the idea of ILC and we rely on it for analysis purpose.

Substituting (7) into (6) gives

$$\mathbf{e}_k = [\mathbf{I} \quad \mathbf{GH}_1]\bar{\mathbf{e}}_{k\ \ 1} + \mathbf{v}_k + \Delta\mathbf{w}_k \tag{8}$$
$$\bar{\mathbf{e}}_k = [\mathbf{I} \quad \mathbf{GH}_1]\bar{\mathbf{e}}_{k\ \ 1} + \Delta\mathbf{w}_k \tag{9}$$

$\mathbf{v}_k$ and $\Delta\mathbf{w}_k$ appear in $\mathbf{e}_k$ without any attenuation. It is a natural result because $\mathbf{v}_k$ and $\Delta\mathbf{w}_k$ are newly entered disturbance at k while $\Delta\mathbf{u}_k$ is calculated based on the previous run information. It can be seen that $\bar{\mathbf{e}}_k$ and, as a consequence, $\Delta\mathbf{u}_{k+1}$ are not a ected by $\mathbf{v}_k$. This implies that ILC based on (6) can keep its integrity rejecting the e ect of the real-time disturbance.

When $\mathbf{v}_k = \Delta\mathbf{w}_k = 0$ and $\|\mathbf{I} \quad \mathbf{GH}_1\| < 1$, $\mathbf{e}_k \to 0$ as $k \to \infty$.

2.3 ILC combined with Real-time Feedback

An ILC algorithm combined with RFC (RFC-ILC) can be expressed as

$$\Delta\mathbf{u}_k = \mathbf{H}_2\mathbf{e}_k \tag{10}$$

To reject the real-time disturbance, $\mathbf{e}_k$ instead of $\bar{\mathbf{e}}_k$ is fed back. For causality, $\mathbf{H}_2$ has a low-triangular structure. Again, the real algorithm may be more complicated than the above. (10) retains the key features of an RFC-ILC algorithm.

Substituting (10) into (6) results in

$$\mathbf{e}_k = [\mathbf{I} + \mathbf{GH}_2]\ \ ^1(\bar{\mathbf{e}}_{k\ \ 1} + \mathbf{v}_k + \Delta\mathbf{w}_k) \tag{11}$$
$$\bar{\mathbf{e}}_k = [\mathbf{I} + \mathbf{GH}_2]\ \ ^1(\bar{\mathbf{e}}_{k\ \ 1} \quad \mathbf{GH}_2\mathbf{v}_k + \Delta\mathbf{w}_k) \tag{12}$$

It can seen that the e ects of $\mathbf{v}_k$ and $\Delta\mathbf{w}_k$ are attenuated in $\mathbf{e}_k$ by the real-time control action. The larger $\mathbf{GH}_2$ is, the more the disturbance is rejected. However, a large $\mathbf{GH}_2$ makes $\bar{\mathbf{e}}_k$ be strongly a ected by $\mathbf{v}_k$, which not only gives an harmful e ect on the ILC track but also deteriorates the performance of $\mathbf{e}_{k+1}$.

3. PROPOSED BATCH CONTROL TECHNIQUE

One of the representative RFC-ILC techniques for batch chemical processes, called BMPC (Lee and Lee, 1997), is based on the updating rule (10), and has the problem discussed in the previous section. On the other hand, a pure ILC technique, called QILC (Lee *et al.*, 2000), is based on (7) and can keep the genuine learning track not being a ected by the real-time disturbance. Bearing the above in mind, we propose a new RFC-ILC formulation where RFC is separated from ILC so that the e ect of the real-time disturbance is blocked from transferring to the learning track.

The proposed technique is designed to perform two-stage calculation: ILC after a run, say it k 1^{th} run, and RFC calculation during the k^{th} run on the basis of the learning input. Detailed ILC and RFC algorithms are constructed by modifying existing QILC and BMPC, respectively.

3.1 Process Modeling

Let us decompose the disturbance into three terms: $\mathbf{w}_k$, $\mathbf{v}_k$, $\mathbf{n}_k$ which refer to the parts that will be rejected by ILC and by RFC, and the measurement noise, respectively.

$$\mathbf{d}_k = \mathbf{w}_k + \mathbf{v}_k + \mathbf{n}_k \tag{13}$$
$$\mathbf{w}_k = \mathbf{w}_{k\ \ 1} + \Delta\mathbf{w}_k.$$

Also we decompose the a control action $\mathbf{u}_k$ into $\bar{\mathbf{u}}_k$ and $\hat{\mathbf{u}}_k$, each of which is responsible for $\Delta\mathbf{w}_k$ and $\mathbf{v}_k$, respectively. With these variables, the process model can be expressed as

$$\begin{aligned}\mathbf{y}_k &= \mathbf{G}\mathbf{u}_k \quad \mathbf{d}_k = \mathbf{G}(\bar{\mathbf{u}}_k + \hat{\mathbf{u}}_k) \quad (\mathbf{w}_k + \mathbf{v}_k + \mathbf{n}_k) \\ &= \underbrace{\underbrace{\mathbf{G}\bar{\mathbf{u}}_k \quad \mathbf{w}_k}_{\bar{\mathbf{y}}_k} + \mathbf{G}\hat{\mathbf{u}}_k \quad \mathbf{v}_k \quad \mathbf{n}_k}_{\hat{\mathbf{y}}_k} \\ &= \bar{\mathbf{y}}_k + \mathbf{G}\hat{\mathbf{u}}_k \quad \mathbf{v}_k \quad \mathbf{n}_k = \hat{\mathbf{y}}_k \quad \mathbf{n}_k \end{aligned} \tag{14}$$

$\bar{\mathbf{u}}_k$ and $\hat{\mathbf{u}}_k$ will be used by ILC and RFC to cancel $\mathbf{w}_k$ and $\mathbf{v}_k$, respectively, while steering $\mathbf{y}_k$ to follow $\mathbf{y}_d$. If perfect disturbance rejection would be made, $\mathbf{y}_k \to \mathbf{y}_d \quad \mathbf{n}_k$ in the end.

Similarly to (6), the following model equation can be derived from (14):

$$\begin{aligned}\bar{\mathbf{e}}_k &= \bar{\mathbf{e}}_{k\ \ 1} \quad \mathbf{G}\Delta\bar{\mathbf{u}}_k + \Delta\mathbf{w}_k \\ \hat{\mathbf{e}}_k &= \bar{\mathbf{e}}_k \quad \mathbf{G}\hat{\mathbf{u}}_k + \mathbf{v}_k \\ \mathbf{e}_k &= \hat{\mathbf{e}}_k + \mathbf{n}_k.\end{aligned} \tag{15}$$

In the above, $\bar{\mathbf{e}}_k = \mathbf{y}_d \quad \bar{\mathbf{y}}_k$ and $\hat{\mathbf{e}}_k = \mathbf{y}_d \quad \hat{\mathbf{y}}_k$, respectively.

3.2 Revised QILC

ILC calculates $\Delta\bar{\mathbf{u}}_k$ instead of $\Delta\mathbf{u}_k$. After the $k \quad 1^{\text{th}}$ run, $\mathbf{e}_{k\ \ 1}$, $\bar{\mathbf{u}}_{k\ \ 1}$, and $\mathbf{u}_{k\ \ 1}$ are available. Then $\Delta\bar{\mathbf{u}}_k$ is determined such that

$$\min_{\Delta\bar{\mathbf{u}}_k} \frac{1}{2}\left\{\|\bar{\mathbf{e}}^T_{k|k\ \ 1}\|^2_{\mathbf{Q}} + \|\Delta\bar{\mathbf{u}}_k\|^2_{\mathbf{R}}\right\} \tag{16}$$
$$\text{subject to } \bar{\mathbf{e}}_{k|k\ \ 1} = \bar{\mathbf{e}}_{k\ \ 1|k\ \ 1} \quad \mathbf{G}\Delta\bar{\mathbf{u}}_k \tag{17}$$
$$\bar{\mathbf{e}}_{k\ \ 1|k\ \ 1} = \bar{\mathbf{e}}_{k\ \ 1|k\ \ 2} + \mathbf{K}(\mathbf{e}_{k\ \ 1} \quad (\bar{\mathbf{e}}_{k\ \ 1|k\ \ 2} \quad \mathbf{G}\hat{\mathbf{u}}_{k\ \ 1}))$$

where $\mathbf{K}$ is the Kalman gain which depends on $\mathbf{R}_{\Delta\mathbf{w}_k}$ and $\mathbf{R}_{\mathbf{v}}$. Inequality constraints on $\bar{\mathbf{e}}_{k|k\ \ 1}$ and $\Delta\bar{\mathbf{u}}_k$ can be incorporated together. The above calculation is repeated after the each run.

3.3 Real-time Predictive Control

Let us de ne the state at $t+1$, which will be regulated by RFC, as

$$\begin{aligned}\hat{\mathbf{e}}_k(t+1) = \hat{\mathbf{e}}_k \quad \text{with} \quad \Delta\bar{u}_k(t+1) = \cdots = 0,\\ \hat{u}_k(t+1) = \cdots = 0,\ v_k(t) = \cdots = 0\\ = \bar{\mathbf{e}}_{k\ \ 1} \quad G(0)(\Delta\bar{u}_k(0) + \hat{u}_k(0)) \quad \cdots\\ G(t)(\Delta\bar{u}_k(t) + \hat{u}_k(t))\\ + \left[\, v_k^T(0)\ \cdots\ v_k^T(t)\ 0\ \cdots \right]^T + \Delta\mathbf{w}_k. \quad (18)\end{aligned}$$

The relationship in the above can be derived from (15) and $G(i)$ represents the i^{th} block column in $\mathbf{G}$. If we write (18) for $\hat{\mathbf{e}}_k(t)$ and take the di erence from $\hat{\mathbf{e}}_k(t)$ while assuming the dynamics of the real-time disturbance as

$$v(t) = \alpha v(t \quad 1) + m(t), \quad (19)$$

the state space equation in time is constructed:

$$\begin{bmatrix} \hat{\mathbf{e}}_k(t+1) \\ v(t+1) \end{bmatrix} = \begin{bmatrix} I & H(t) \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} \hat{\mathbf{e}}_k(t) \\ v(t) \end{bmatrix} \quad (20)$$
$$\begin{bmatrix} G(t) \\ 0 \end{bmatrix} (\Delta\bar{u}_k(t) + \hat{u}_k(t)) + \begin{bmatrix} 0 \\ I \end{bmatrix} m(t)$$

$$e_k(t) = \left[\, H^T(t)\ 0 \,\right] \begin{bmatrix} \hat{\mathbf{e}}_k(t) \\ v(t) \end{bmatrix} + n(t) \quad (21)$$

$$\text{with} \quad \begin{bmatrix} \hat{\mathbf{e}}_k(0) \\ v(0) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{e}}_{k\ \ 1} + \Delta\mathbf{w}_k \\ v_0 \end{bmatrix} \quad (22)$$

where $H(t)$ is a zero matrix except I at the t^{th} block column such that

$$H(t) = [\overbrace{0\ \cdots\ 0}^{t\ \ 1\ \text{cols}}\ I\ 0\ \cdots\ 0]^T. \quad (23)$$

For simplicity, let's de ne

$$\underline{\Delta}u_k(t) = \Delta\bar{u}_k(t) + \hat{u}_k(t) = u_k(t) \quad \bar{u}_{k\ \ 1}(t) \quad (24)$$

The prediction equation can be readily derived from (20). Let $\left[\hat{\mathbf{e}}_k^T(t+m|t)\ v^T(t+m|t)\right]^T$ be the prediction of the state made at time t when there are m future control moves. Then we have

$$\begin{bmatrix} \hat{\mathbf{e}}_k(t+m|t) \\ v(t+m|t) \end{bmatrix} = \begin{bmatrix} I & \sum_{j=0}^{m\ \ 1} H(t+j) \\ 0 & \alpha I \end{bmatrix} \begin{bmatrix} \hat{\mathbf{e}}_k(t|t) \\ v(t|t) \end{bmatrix}$$
$$\begin{bmatrix} \mathbf{G}^m(t) \\ 0 \end{bmatrix} \underline{\Delta}\mathbf{u}_k^m(t) \quad (25)$$

where

$$\begin{aligned}\mathbf{G}^m(t) &= \left[\, G(t),\ \cdots,\ G(t+m \quad 1) \,\right],\\ \underline{\Delta}\mathbf{u}_k^m(t) &= \begin{bmatrix} \underline{\Delta}u_k(t) \\ \vdots \\ \underline{\Delta}u_k(t+m \quad 1) \end{bmatrix}\end{aligned} \quad (26)$$

and $\left[\hat{\mathbf{e}}_k^T(t|t)\ v^T(t|t)\right]^T$ is the state estimate by the Kalman lter applied to (20)-(22).

$\underline{\Delta}\mathbf{u}_k(t)$ is calculated to minimize

$$J = \frac{1}{2}\left\{\|\hat{\mathbf{e}}_k(t+m|t)\|_\Gamma^2 + \|\underline{\Delta}\mathbf{u}_k^m(t)\|_\Lambda^2\right\} \quad (27)$$

Inequality constraints can be imposed on the output and input variables.

Note that that calculation of $\underline{\Delta}u_k(t)$ is equivalent to calculation of $\hat{u}_k(t)$ since $\bar{u}_{k\ \ 1}(t)$ is known. After the batch run, $\hat{\mathbf{u}}_k$ is available and $\Delta\bar{\mathbf{u}}_{k+1}$ can be computed by QILC.

ILC in the proposed technique is based on the updating rule in (7). Hence the learning is basically una ected by the run-independent disturbance even under the active feedback action by the predictive control.

3.4 Tuning Guideline

It is thought that tuning through the noise covariance matrices is more transparent than through the quadratic cost weighting matrices in the proposed controller. In fact, it is well known that the covariance matrices and the weighting matrices have symmetric e ects on the controller performance (Lee *et al.*, 2000). In the constrained case, however, the input penalty in the quadratic cost loses its meaning when the input is stuck on the constraint boundary, while the Kalman gain can still function as an e ective tuning knob. In this respect, it is suggested to x $\mathbf{Q}$ and Γ as scaling matrices and set $\mathbf{R}$ and Λ as small positive de nite matrices only for regularization, and to use $\mathbf{R}_{\Delta w}$, $\mathbf{R}_v$ (determined by $\mathbf{R}_m$ according to (19), and $\mathbf{R}_n$ as the tuning factors. Their e ects on the respective controllers are rather obvious from the nature of the Kalman lter. If $\mathbf{R}_{\Delta w}$ is given to be large in relation to $\mathbf{R}_v + \mathbf{R}_n$ (note that $\mathbf{v}_k + \mathbf{n}_k$ in (13) is equivalent to $\mathbf{v}_k$ in (4)), the run-independent disturbance is weakly ltered and has a strong in uence on the ILC performance. Therefore such a choice should be made when the run-independent disturbance is not large. By the similar reasoning, the behavior of RFC is determined by the ratio $\mathbf{R}_{\Delta w} + \mathbf{R}_v$ to $\mathbf{R}_n$. When the ratio is large, real-time control is enhanced.

4. NUMERICAL ILLUSTRATIONS

The performance of the proposed algorithm is examined for two numerical processes, a linear single-input single-output (SISO) batch process and a semi-batch reactor system with series-parallel reactions (Chin *et al.*, 2000).

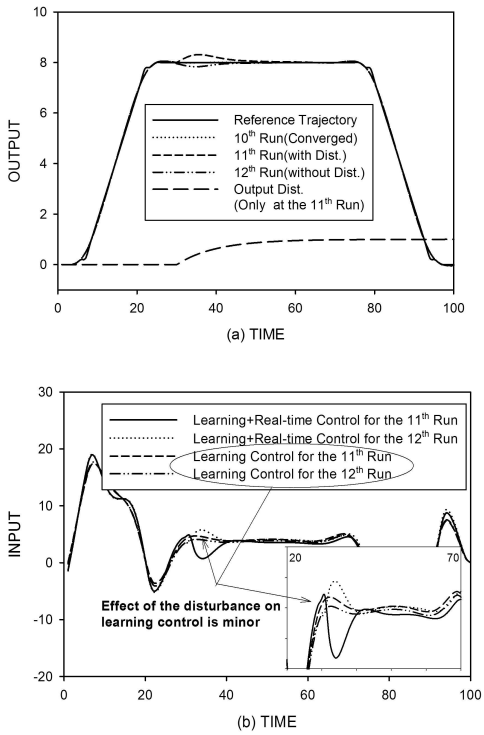


Fig. 1. (a) Controlled variables (b) Manipulated variables for the 11^{th} and 12^{th} runs under a run-independent disturbance.

4.1 Linear SISO System

The plant and nominal model are the sampled-data systems (sampling interval=1) of the following transfer functions, respectively:

$$G^p = \frac{2.5}{300s^2 + 35s + 1},\ G^m = \frac{1.5}{270s^2 + 33s + 1} \quad (28)$$

It is assumed that a run-independent disturbance, a step response of a low pass filter, enters from $t = 31$ only at the 11^{th} run.

4.1.1. Results and Discussion Figure 1 shows the performance of the proposed control technique. It can be seen that $\bar{u}_{12}(t)$ is only slightly influenced by the run-independent disturbance although the disturbance is aggressively rejected by the real-time predictive control yielding largely changing $u_{11}(t)$. The consequence is that the learning process can be continued with only a minor interrupt. Such a performance cannot be achieved by the existing RFC-ILC techniques.

4.2 Semi-Batch Reactor

The jacketed semi-batch reactor model in (Chin *et al.*, 2000), where the following reaction takes place

$$A + B \overset{k_1}{\rightarrow} C$$

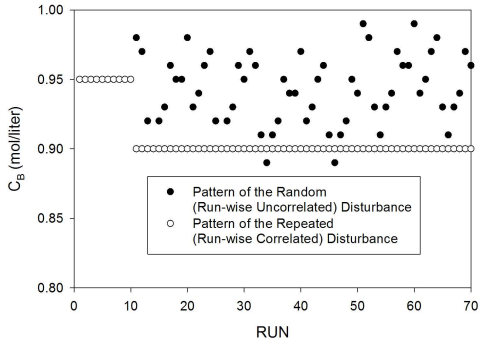


Fig. 2. Two different disturbance scenarios.

$$B + C \overset{k_2}{\rightarrow} D \quad (29)$$

is revisited in this example. A is charged initially and the heat-up is followed until B starts to be fed at $t = 31$ min. The reaction commences at this point and continues until the batch terminal time of $t_f = 100$ min. During this period, A is sampled at every 10 min for concentration measurement. The desired product is C and the main objective is to maintain the final yield of C at 42 mol. We considered two manipulated variables: jacket temperature $T_j(t)$ and flow rate of B, $Q_B(t)$ where the following constraints are imposed:

$$20^{\text{o}}\text{C} \le T_j(t) \le 45^{\text{o}}\text{C} \quad (30)$$
$$0.5\ (\text{liter/min}) \le Q_B(t) \le 1.5\ (\text{liter/min})$$

The sample time for control was chosen to be 1 min. In this example, two different disturbance patterns in C_B (concentration of feed B) are assumed as shown in Figure 2. In the first case, C_B changes randomly around 0.95 (mol/l) from the 11^{st} run. In the second case,C_B is decreased from 0.95 (mol/l) to 0.9 (mol/l) at the 11^{th} run and kept at 0.9 (mol/l) thereafter.

4.2.1. Results and Discussion Figure 3 shows a result for the first disturbance scenario (run-independent disturbance). One can see that $\bar{u}_k(t)$ is almost uninfluenced by the disturbance and remains on the already-converged input trajectories. In Figure 4, a result for the repeated disturbance is given. It can be observed that $\bar{u}_k(t)$ changes and converges to new profiles that can perfect reject the repeated disturbance. Although not shown here due to limited space, the performance of quality (final yield of C) control as well temperature tracking control were found to be quite satisfactory for both disturbance scenarios.

5. CONCLUSIONS

We proposed a new learning control methodology to handle two different types of disturbances, run-wise persisting and uncorrelated disturbances

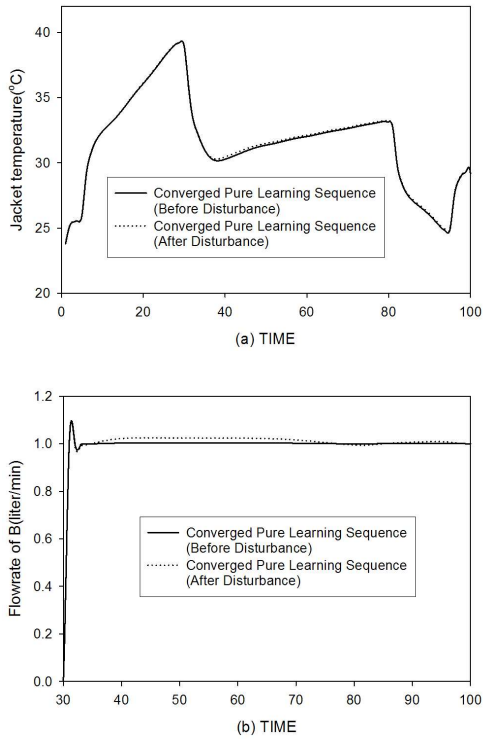


Fig. 3. (a) Jacket temperature (b) feed owrate of B under the run-independent disturbance.

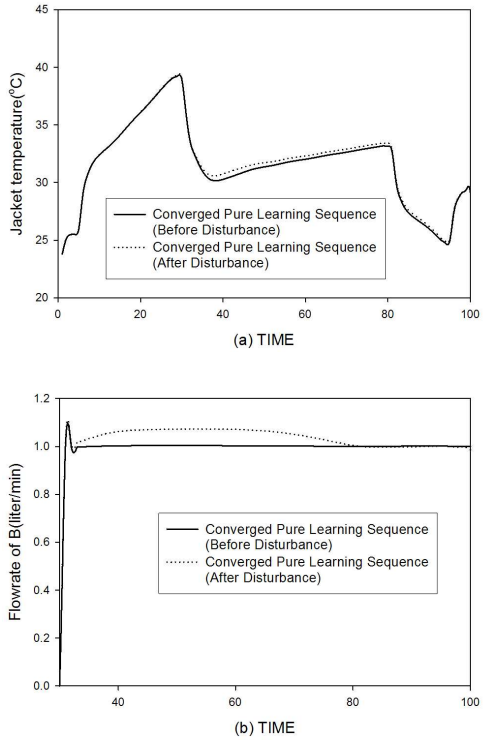


Fig. 4. (a) Jacket temperature (b) feed owrate of B under the run-wise repeated disturbance.

separately with a simple tuning guideline. The present BMPC or other combined iterative learning control (ILC) and real-time feedback control (RFC) methods share a problem that an excessive input movement by a large real-time disturbance is transferred to the next run as a feedforward input signal, which leads to deterioration of the learning performance. To solve this problem, we have devised a two-stage algorithm, RFC during a batch run to ght against run-wise uncorrelated disturbance and then ILC after the batch run for input update only by the run-wise persisting disturbance. The proposed control algorithm is based on the earlier study on BMPC and QILC for the inheritance of their advantages.

Numerical studies reveal that the proposed technique works as anticipated overcoming the problem of existing batch control methods.

To the authors' knowledge, the present paper is the rst achievement that correctly deals with the disturbance rejection problem in batch process control. The two-stage technique based on QILC and BMPC has been given as a prototypical technique to realize the idea.

REFERENCES

Arimoto, S., S. Kawamura and F. Miyazaki (1984). Bettering operation of robots by learning. *J. Robotic Syst.* **1(1)**, 123.

Bien, Z. and K. M. Huh (1989). Higher-order iterative learning control algorithm. *IEE Proc. Part D* **136(3)**, 105–112.

Chin, I. S., K. S. Lee and J. H. Lee (2000). A technique for integrated quality control, pro le control and constraint handling for batch processes. *Ind. & Eng. Chem.* **39**, 693–705.

Good, R., J. Hahn, T. Edison and S. J. Qin (2002). A run-to-run control approach to drug dosage adjustment for long-term drug therapy. *Submitted to Control Engineering Practice.*

Lee, K. S. and J. H. Lee (1997). Model predictive control for nonlinear batch processes with asymptotically perfect tracking. *Com. & Chem. Eng.* **21**, 873–879.

Lee, K. S., J. H. Lee and W. C. Kim (2000). Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica* **36**, 641–657.

Lee, K. S., J. Lee, I. S. Chin, J. Choi and J. H. Lee (2001). Control of wafer temperature uniformity in rapid thermal processing using an optimal iterative learning control technique. *Ind. Eng. Chem. Res.* **40**, 1661–1672.

Lee, K. S., S. H. Bang and K. S. Chang (1994). Feedback-assisted iterative learning control based on an inverse process model. *Journal of Process Control* **4**, 77.

Qin, S. J., G. Scheid and T. Riley (2002). Run-to-run optimization, monitoring, and control on a rapid thermal processor. *Submitted to JVST-B.*

RESULTS ANALYSIS IN A CONSTRAINED REAL-TIME OPTIMIZATION (RTO) SYSTEM

W. S. Yip and T. E. Marlin

Department of Chemical Engineering, McMaster University, Hamilton, ON L8S 4L7, Canada

Abstract: This paper presents a new results analysis strategy for uncertain real-time optimization (RTO) systems. The key contributions of this paper are in 1) developing a mathematical formulation describing the confidence region of a partially constrained optimum and 2) applying that formulation in deciding whether the optimizer results should be implemented. The confidence region of the constrained optimum is determined from the confidence region of the unconstrained optimum, when the inequality constraints are not considered. The confidence region of the unconstrained optimum is mapped to the feasible region defined by the inequality constraints to obtain the confidence region of the constrained optimum. This mapping is developed by minimizing the profit loss between the unconstrained and constrained optima. The resulting confidence region of the constrained optimum is used in optimization results analysis; the confidence regions of successive predicted optima are compared to decide if the difference between two optima is statistically significant. The comparison is made by calculating the significance level at which the two confidence regions just overlap. If the value is small, large portions of two 95% confidence regions overlap, and the optimizer result will not be implemented. The new results analysis approach is applied to the Williams-Otto reactor case study with controlled inequality constraints, where it significantly reduced plant variability, while concurrently increasing the operating profit.

Keywords: Real-time optimisation, results analysis, constrained optimum

1. INTRODUCTION

Real-time operations optimization (RTO) can improve the operating profit of chemical plants by tracking the changing optimum due to disturbances such as changes in plant performance and external state variables. This paper focuses on the model-based real-time operations optimization using steady-state models, in which the RTO execution period is much longer than the feedback dynamics. In this situation, a steady-state model is sufficient for economic optimization.

The main elements in the RTO loop [Marlin and Hrymak, 1997] consist of the model updater, model-based optimizer, results analysis and process control as shown in Figure 1. Real-time measurements ($\mathbf{z}$) are taken from the plant, checked for reliability and low pass filtered. Then, the process parameters ($\boldsymbol{\beta}$) are estimated using the data in the model updater. The estimated parameters are then sent to the optimizer, in which model-based optimization is performed. The optimizer results are analyzed in results analysis [Miletic and Marlin, 1998] before being transmitted to the process controllers. Only significant changes in optimization variables are forwarded to the process controllers for implementation. The new setpoint can be determined by trading off the change in profit and the size of the change of the operating variables [Ronholm and Marlin, 2002] to make the transition more gradual.

The performance of an RTO system is measured by two terms: 1) *offset* between the plant optimum and noise-free model prediction, and 2) *variability* of the prediction. Offset is caused by the structural mismatch between the plant and the model and the errors in the parameter values. Variability is caused

by high frequency disturbances and measurement noise propagating in the RTO loop. Offset and variability can have a significant impact on the operating profit, and a small offset and variability is desirable in profit tracking.

This paper focuses on improving the RTO performance by reducing the variability of the manipulated variables in a partially constrained RTO system. The RTO results are influenced by high frequency variation and should be evaluated with respect to the common cause variability. In results analysis [Miletic and Marlin, 1998], the newly predicted (uncertain) optimum is compared with the previous (uncertain) results in deciding whether the new result shall be implemented or not. In previous work, it was assumed that the active set of inequality constraints remains unchanged when the parameters are perturbed. Therefore, the covariance matrix of the predicted optimum can be estimated by linear sensitivity analysis and a statistical test can be formulated as a Hotelling T^2 test. In this paper, the assumption of a constant active set will be relaxed and a new strategy of results analysis will be developed to handle the possible change in the active set of inequality constraints.

The outline of this paper is as follows. The mathematical formulation to describe the confidence region of the constrained optimum is first developed. The new strategy of results analysis using the developed formulation for the uncertainty of the constrained optimum is then presented. Finally, the proposed approach is applied to the Williams-Otto reactor system to investigate if the results analysis can reduce the unnecessary plant movement responding to high frequency disturbances.

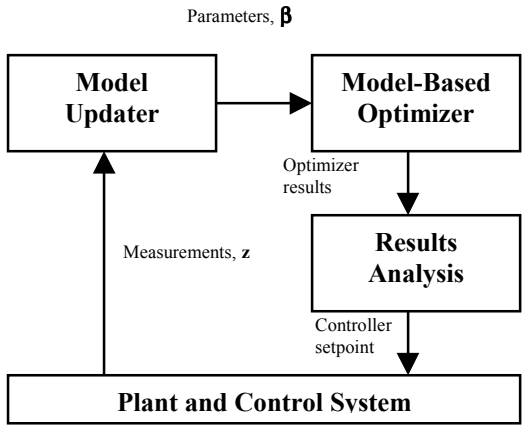


Fig. 1 Closed-loop RTO system

2. MATHEMATICAL FORMULATION FOR THE UNCERTAINTY OF THE CONSTRAINED OPTIMUM

In this section, the method to estimate the confidence region of the constrained optimum is presented. The optimization problem considered in this paper is formulated as follows

$$\underset{\mathbf{x},\,\mathbf{u}}{\text{Maximize}} \quad P(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}) \tag{1}$$

$$\text{Subject to} \quad \mathbf{h}(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}) = \mathbf{0}$$
$$\mathbf{w}(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}) \le \mathbf{0}$$

where $\mathbf{x}$ is the vector of decision variables implemented to the plant, $\mathbf{u}$ is the vector of dependent variables, $\mathbf{h}$ and $\mathbf{w}$ are the equality and inequality constraints respectively and P is the objective function. The vector of parameters, $\boldsymbol{\beta}$, is estimated from the updater, and its uncertainty is described by the confidence region given in (2)

$$\left(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\right)^{T}\mathbf{Q}_{\boldsymbol{\beta}}^{-1}\left(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\right)\le\chi^{2}(\alpha,\nu_{\beta}) \tag{2}$$

where $\hat{\boldsymbol{\beta}}$ is the nominal value of the estimated parameters, $\mathbf{Q}_{\beta}$ is the covariance matrix of the estimated parameters, χ^2 is the Chi-square distribution, α% is the level of confidence and ν_β is the degrees of freedom which is equal to the number of the estimated parameters, assuming the covariance matrix is known [Anderson, 1984]. In this work, a possible change in the active set of the inequality constraints, $\mathbf{w}$, may occur for the anticipated uncertainty of $\boldsymbol{\beta}$ given in (2).

The confidence region of the constrained optimum, $\mathbf{x}_c^*$, which is the solution of (1), is obtained by "mapping" the confidence region $\mathbf{x}_u^*$ to the region defined by the inequality constraints as shown in Figure 2. The unconstrained optimum, $\mathbf{x}_u^*$, is the solution of the following optimization problem, which is the original problem (1) without the inequality constraints.

$$\underset{\mathbf{x},\,\mathbf{u}}{\text{Maximize}} \quad P(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}) \tag{3}$$

$$\text{Subject to} \quad \mathbf{h}(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}) = \mathbf{0}$$

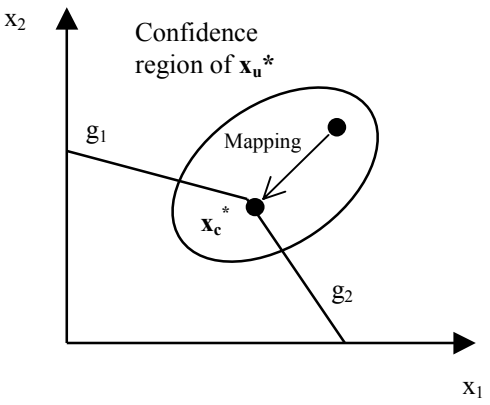


Fig. 2 Transformation of the uncertainty from the constant active set to the feasible region

The confidence region of $\mathbf{x}_u^*$ can be determined by linear sensitivity analysis [Fiacco, 1983] of problem (3) because the active set of inequality constraints remains constant. All linearizations are performed at the nominal value of the estimated parameter, $\hat{\boldsymbol{\beta}}$, and the unconstrained optimum, $\hat{\mathbf{x}}_u^*$, estimated from $\hat{\boldsymbol{\beta}}$. The covariance matrix of $\mathbf{x}_u^*$ is given in (4).

$$\mathbf{Q}_{\mathbf{x}_u^*} = \left(\frac{d\mathbf{x}_u^*}{d\boldsymbol{\beta}}\right)\mathbf{Q}_{\beta}\left(\frac{d\mathbf{x}_u^*}{d\boldsymbol{\beta}}\right)^T \qquad (4)$$

The procedure in calculating the sensitivity matrix $\frac{d\mathbf{x}_u^*}{d\boldsymbol{\beta}}$ for problem (3) is discussed in Wolbert, et al. (1994). The confidence region of $\mathbf{x}_u^*$ is given in (5)

$$\left(\mathbf{x}_u^* - \hat{\mathbf{x}}_u^*\right)^T \mathbf{Q}_{\mathbf{x}_u^*}{}^{-1}\left(\mathbf{x}_u^* - \hat{\mathbf{x}}_u^*\right) \le \chi^2(\alpha, \nu_X) \qquad (5)$$

where ν_x is the degrees of freedom in (3) which is equal to the dimension of $\mathbf{x}$.

The confidence region of $\mathbf{x}_c^*$ can be derived by mapping every point inside the confidence region of $\mathbf{x}_u^*$ given in (5) to the feasible region defined by the inequality constraints in the reduced space. By eliminating $\mathbf{u}$ using $\mathbf{h}(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}) = \mathbf{0}$, Equation (1) can be written as follows

$$\underset{\mathbf{x}}{\text{Maximize}} \quad P_r(\mathbf{x}, \boldsymbol{\beta}) \qquad (6)$$

$$\text{Subject to} \quad \mathbf{g}(\mathbf{x}, \boldsymbol{\beta}) \le \mathbf{0}$$

where P_r and $\mathbf{g}$ are the objective function and inequality constraints in the reduced space. The solution of (6) is exactly the same as $\mathbf{x}_c^*$ if we can analytically solve for $\mathbf{u}$ from $\mathbf{h}(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}) = \mathbf{0}$. If we must linearize $\mathbf{h}(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}) = \mathbf{0}$ to eliminate $\mathbf{u}$, Equation (6) is an approximation of (1).

As shown in Figure 2, the collection of the points after the mapping in the region bounded by $\mathbf{g}(\mathbf{x}, \boldsymbol{\beta}) \le \mathbf{0}$ defines the confidence region of the constrained optimum. For a given value of $\boldsymbol{\beta}$ inside the confidence region defined in (2), the unconstrained optimum $\mathbf{x}_u^*$ can be estimated from the linear approximation as follows.

$$\mathbf{x}_u^* - \hat{\mathbf{x}}_u^* = \frac{d\mathbf{x}_u^*}{d\boldsymbol{\beta}}\left(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right) \qquad (7)$$

For every value of $\boldsymbol{\beta}$, the difference in profit at $\mathbf{x}_u^*$ and $\mathbf{x}_c^*$ should be minimum. Therefore, (6) can be expressed without changing the result [$P_r(\mathbf{x}_u^*, \boldsymbol{\beta})$ is a constant] as follows

$$\underset{\mathbf{x}}{\text{Minimize}} \; P_r(\mathbf{x}_u^*, \boldsymbol{\beta}) - P_r(\mathbf{x}, \boldsymbol{\beta}) \qquad (8)$$

$$\text{Subject to} \quad \mathbf{g}(\mathbf{x}, \boldsymbol{\beta}) \le \mathbf{0}$$

and $\mathbf{x}_c^*$ is the optimum solution of (8). Furthermore, when $P_r(\mathbf{x}, \boldsymbol{\beta})$ can be approximated by expanding the profit function using Taylor's series around the point $\hat{\mathbf{x}}_u^*$, Equation (8) can be re-written as follows.

$$\underset{\mathbf{x}}{\text{Minimize}} \; -\left(\mathbf{x} - \mathbf{x}_u^*\right)^T \nabla_r{}^2 P\Big|_{\hat{\mathbf{x}}_u^*}\left(\mathbf{x} - \mathbf{x}_u^*\right) \qquad (9)$$

$$\text{Subject to} \quad \mathbf{g}(\mathbf{x}, \boldsymbol{\beta}) \le \mathbf{0}$$

The reduced Hessian matrix can be obtained when calculating the sensitivity matrix in (4) using Wolbert's approach [1994]. For any point $\boldsymbol{\beta}$ sampled from the confidence region in (2), $\mathbf{x}_c^*$ can be estimated by solving (7) and (9). We can estimate the uncertain ranges of $\mathbf{x}_c^*$ by solving (7) and (9) for different values of $\boldsymbol{\beta}$ sampled from (2). If the inequality constraints are linear(ized), (9) becomes a quadratic programming (QP) problem that we can solve efficiently.

The confidence region obtained by solving (7) and (9) for a sample of $\boldsymbol{\beta}$ values is characterized by a collection of data points. In results analysis, it is difficult to evaluate two uncertain optimizer results from two sets of data points, since this is a bilevel optimization problem. Therefore, (9) is reformulated to a system of algebraic equations using Karash-Kuhn-Tucker (KKT) conditions so that we can reformulate the results analysis easily. By taking the KKT conditions of (9), we can obtain the following system of equations that are equivalent to equations (7) and (9), along with a restriction on $\boldsymbol{\beta}$ being within its approximate α % confidence region.

$$-2\nabla_r{}^2 P\Big|_{\hat{\mathbf{x}}_u^*}\left(\mathbf{x} - \mathbf{x}_u^*\right) + \nabla_{\mathbf{x}}\mathbf{g}(\mathbf{x}, \boldsymbol{\beta})^T \boldsymbol{\mu} = \mathbf{0} \qquad (10a)$$

$$\mathbf{g}(\mathbf{x}, \boldsymbol{\beta}) \le \mathbf{0} \qquad (10b)$$

$$\mu_i g_i(\mathbf{x}, \boldsymbol{\beta}) = 0 \qquad (10c)$$

$$\mu_i \ge 0 \qquad (10d)$$

$$\left(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right)^T \mathbf{Q}_{\boldsymbol{\beta}}{}^{-1}\left(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right) \le \chi^2(\alpha, \nu_\beta) \qquad (10e)$$

$$\mathbf{x}_u^* - \hat{\mathbf{x}}_u^* = \frac{d\mathbf{x}_u^*}{d\boldsymbol{\beta}}\left(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right) \qquad (10f)$$

where i denotes the i^{th} inequality constraint. Equations (10e) and (10f) define the uncertainty of $\boldsymbol{\beta}$ and the unconstrained optimum, $\mathbf{x}_u^*$. Equations (10a) – (10d) defines the mapping which transforms the uncertainty of $\mathbf{x}_u^*$ in the reduced space to the uncertainty of $\mathbf{x}_c^*$ in the region defined by $\mathbf{g}(\mathbf{x}, \boldsymbol{\beta}) \le \mathbf{0}$. Because of the change in the active set, the

sensitivity matrix of $\mathbf{x}_c^*$ with respect to $\boldsymbol{\beta}$ (i.e. $\frac{d\mathbf{x}_c^*}{d\boldsymbol{\beta}}$) is not defined over the entire region of interest. The change in the active set is represented by the complementarity equation in (10c). Equations (10a) – (10f) describing the α% of confidence region of $\mathbf{x}_c^*$ will be used for results analysis in RTO systems.

3. RESULTS ANALYSIS IN REAL-TIME OPTIMIZATION

In this section, a new results analysis strategy to handle possible changes in the active set of inequality constraints is presented. The control structure has to be considered in results analysis. In this paper, we focus on the case in which the inequality constraints are the bounds on the decision variables, which are final elements or setpoints regulated by controllers at a much higher frequency than the RTO loop. In this situation, the inequality constraints are not a function of $\boldsymbol{\beta}$. Therefore, Equations (10a) – (10f) can be simplified further as follows

$$-2\nabla_r{}^2 P\Big|_{\hat{\mathbf{x}}_u^*}\left(\mathbf{x}-\mathbf{x}_u^*\right)+\nabla_x \mathbf{g}(\mathbf{x})^T\boldsymbol{\mu}=\mathbf{0} \quad (11a)$$

$$\mathbf{g}(\mathbf{x}) \le \mathbf{0} \quad (11b)$$

$$\mu_i g_i(\mathbf{x}) = 0 \quad (11c)$$

$$\mu_i \ge 0 \quad (11d)$$

$$\left(\mathbf{x}_u^*-\hat{\mathbf{x}}_u^*\right)^T \mathbf{Q}_{\mathbf{x}_u^*}{}^{-1}\left(\mathbf{x}_u^*-\hat{\mathbf{x}}_u^*\right)\le \chi^2(\alpha,\nu_x) \quad (11e)$$

where (10e) and (10f) are combined to form (11e) and $\mathbf{Q}_{\mathbf{x}_u^*}$ can be calculated from (4). Equations (11a) – (11e) defines the α% confidence region of the constrained optimum which will be used in results analysis.

The proposed strategy of results analysis compares the overlapping of two confidence regions of the optima predicted in successive RTO executions. The strategy is shown in Figure 3. The level of confidence, α%, that two confidence regions start overlapping is first estimated. If α is small ($\alpha \le \alpha_0$), we expect that large portions of 95% confidence regions of two uncertain optima overlap. Therefore, we conclude that the difference between two uncertain optima is due to common cause variability, and the new setpoint will not be implemented. If α is large ($\alpha > \alpha_0$), only small portions of 95% confidence regions overlap or two 95% confidence regions are even disjoint when α% is larger than 95%. We conclude that the new setpoint should be implemented, because the difference between two uncertain optima is due to non-stationary disturbances.

The confidence level, α%, at which two confidence regions just overlap can be determined by solving an optimization problem. When two confidence regions just overlap, there exists a value of $\mathbf{x}$ that satisfies the system of equations (11a) – (11e) for the last implemented and current RTO executions. Therefore, α can be calculated from the parameter, c, in (12).

$$\underset{\mathbf{x}}{\text{Minimize}}\ c \quad (12)$$

Subject to

$$-2\nabla_r{}^2 P\Big|_{\hat{\mathbf{x}}_{uj}^*}\left(\mathbf{x}-\mathbf{x}_{uj}^*\right)+\nabla_x \mathbf{g}(\mathbf{x})^T\boldsymbol{\mu}_j=\mathbf{0}$$

$$\mathbf{g}(\mathbf{x}) \le \mathbf{0}$$

$$\mu_{ij} g_i(\mathbf{x}) = 0$$

$$\mu_{ij} \ge 0$$

$$\left(\mathbf{x}_{uj}^*-\hat{\mathbf{x}}_{uj}^*\right)^T \mathbf{Q}_{\mathbf{x}_{uj}^*}{}^{-1}\left(\mathbf{x}_{uj}^*-\hat{\mathbf{x}}_{uj}^*\right)\le c$$

and α can be found from the Chi-square distribution table as $c=\chi^2(\alpha,\nu_x)$. In (12), j = 1, 2 which denotes the previous and current RTO results. The value, α, is compared to a pre-specified value, α_0. If α is larger than α_0, the new setpoint will be implemented in the plant.

The parameter, α_0, is the tuning parameter for the results analysis for trading off variability and tracking. If α_0 is chosen to be 0%, results analysis will be turned off and all the variability will be transmitted to the plant. If α_0 is chosen to be 100%, there will be no variability, but the RTO system cannot track the changing optimum. Therefore, the designer has to choose a value of α between 0% and 100% to achieve an appropriate trade-off between tracking and variability.

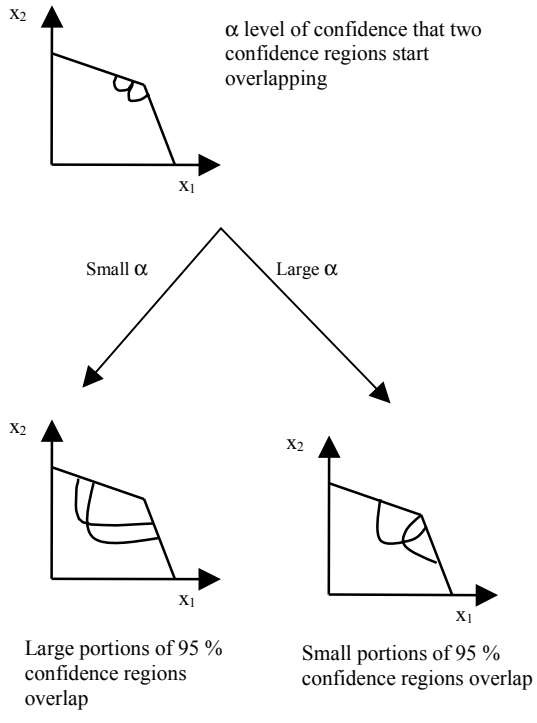
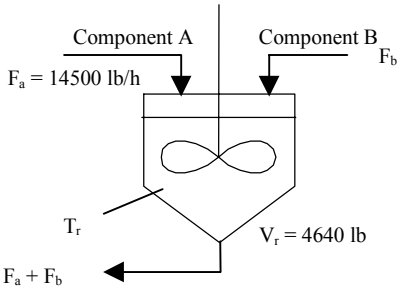


Fig. 3 Strategy of results analysis

The proposed results analysis strategy compares two approximate confidence regions linearized at the current and newly predicted optima. This strategy is appropriate when the current and newly predicted optima are "close" to each other, so that using linearized confidence regions for results analysis is a good approximation. In typical RTO applications, the plant model is optimized subject to a trust region to avoid a large plant movement in a single step. Since large plant moves are prevented in RTO, results analysis using linear confidence regions is appropriate.

4. WILLIAMS-OTTO REACTOR CASE STUDY

The proposed approach is applied to the simulated Williams-Otto reactor system described in Yip and Marlin (2002) as shown in Figure 4. It is assumed that there is no structural mismatch between the plant and the model used in RTO, but substantial parametric uncertainty exists. The optimization variables are the setpoint of the feed flow rate of B (F_b) and reactor temperature (T_r).



$$A + B \rightarrow C \qquad k_1 = A_1 e^{-B_1/T_r}$$

$$C + B \rightarrow P + E \qquad k_2 = A_2 e^{-B_2/T_r}$$

$$P + C \rightarrow G \qquad k_3 = A_3 e^{-B_3/T_r}$$

Fig. 4 Williams-Otto reactor system

The parameters selected for updating are the frequency factors, $\boldsymbol{\beta}^T = [A_1\ A_2\ A_3]$, using a single data set for updating. The measured variables are the reactor volume and temperature, feed flow rates of A and B, and all the compositions. Zero mean white noise is added to the process variables to simulate the measurement errors. The standard deviations for measurement noise are 1% for flows, 2% for reactor volume, 3% for composition and 3.3 R for temperature.

The inequality constraints in the optimizer are the maximum flow rate of feed B and the maximum allowable reactor temperature. There are no inequality constraints in model updating. Therefore, the covariance matrix for the estimated parameter in Equation (2) can be determined by linear sensitivity analysis. The confidence region of the optimizer results is determined from Equation (9) or Equations (11a) – (11e), where the covariance matrix of $\mathbf{x}_u^*$ is related to the covariance matrix of $\boldsymbol{\beta}$ by Equation (4). The approximate 95% confidence region for the constrained optimum gives a good approximation for the uncertain constrained optimum, as shown in Figure 5. Samples of $\boldsymbol{\beta}$ were used in the 500 Monte-Carlo nonlinear optimizations. The approximate 95% confidence region for the constrained $\mathbf{x}_c^*$ is bounded by solid lines on the constraints and the solid curve inside the feasible region. Most of the data points obtained from nonlinear optimizations are inside the approximated 95 % confidence region.

The performance of the RTO system with and without results analysis is shown in Figure 6. In this case study, 100 closed-loop RTO calculations were simulated and a 50% step increase in A_1 occurred at the 50th RTO execution. Without results analysis, there are excessive plant movements due to the propagation of measurement noise. When the results analysis in (12) is implemented, unnecessary plant movements have been significantly reduced as shown in Figure 6.

The performance of the strategy for one value of the tuning parameter (α_0) is shown in Figure 6. When α_0 is equal to 50 %, an increase in total operating profit is achieved because of the reduction of the plant movements. Plant movements can further be reduced by choosing a larger value of α_0. However, the total profit achieved may be lower because the RTO system becomes less effective in profit tracking. When α_0 was chosen to be 0.7, the total profit achieved was \$106370, which was smaller than profit attained when there was no results analysis. Therefore, the designer has to trade off tracking and variability when tuning the results analysis.

The results analysis optimization problem in (12) may have multiple solutions because of the complementarity equations in the constraints, which are non-convex. We need to make sure that the solution of (12) is the global minimum for comparing with α_0. In the Williams-Otto reactor case study, after the value of α had been obtained, the α% confidence regions of two successive predicted optima were plotted to make sure that the solution was reasonable. This can be done because the optimization problem has a dimension of two. In large dimensional problem, we may need global optimization or re-formulation to obtain the global minimum of (12).

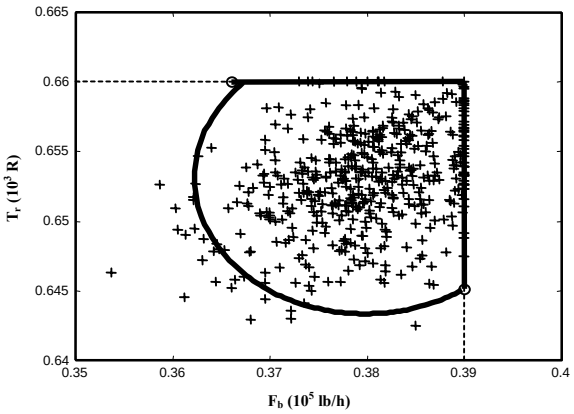


Fig. 5 Comparison of the approximated 95 % confidence region for the constrained optimum (bounded by solid lines on the constraints and solid curve inside the feasible region) with nonlinear optimization for sampled values of β (+).

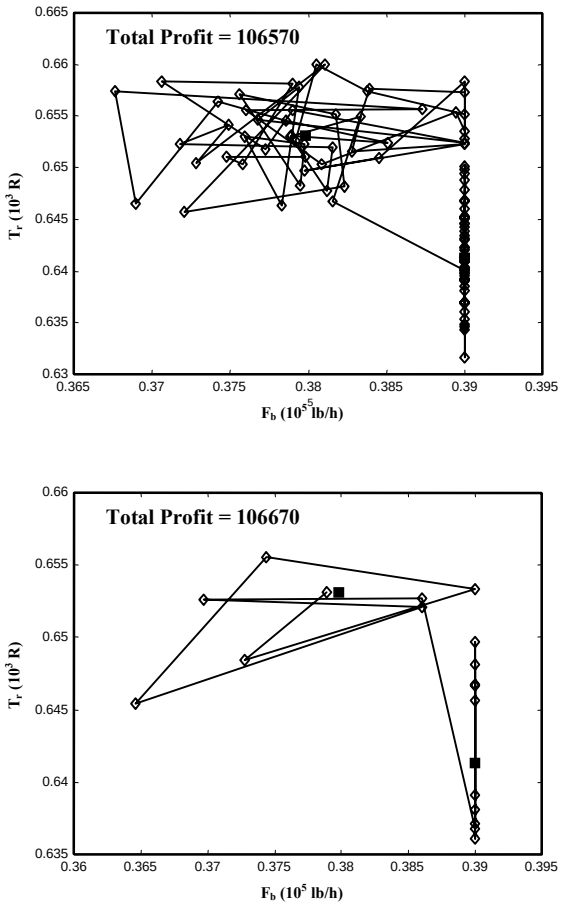


Fig. 6 RTO performance with and without results analysis for tracking the disturbance change of +50% change in A_1. Top: No results analysis, bottom: α_0 equal to 0.5. (◊ : setpoint implemented, ■ : true optimum before and after the disturbance)

5. CONCLUSIONS

A new results analysis strategy for a constrained RTO system has been proposed in this paper. The uncertainty of the constrained optimum, which represents the common cause variability, is determined by mapping the confidence region of the unconstrained optimum to feasible region. The mapping equations contain the complementarity equations, which model the change in the active set of inequality constraints for the anticipated uncertainty in the parameters. The mapping equations are used in the results analysis to compare two uncertain optima. The strategy of results analysis involves determining the level of confidence that two confidence regions start overlapping. The level of confidence is compared to a pre-specified value to determine if the optimizer result shall be implemented or not.

The proposed results analysis strategy has been successfully applied to the simulated Williams-Otto reactor case study. Plant movements responding to measurement noise have been significantly reduced and a higher operating profit can be achieved for a well-chosen tuning parameter.

6. REFERENCES

Anderson, T. W. (1984). *An Introduction to Multivariate Statistical Analysis*. Wiley, NY.

Fiacco, A. V. (1983). *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press, NY.

Marlin, T. E. and Hrymak, A. N. (1997). Real-Time Optimization of Continuous Processes. *Fifth International Conference on Chemical Process Control. AIChE Symp*., No. 316, 156, 1997.

Miletic, I. P. and Marlin, T. E. (1991). On-Line Statistical Results Analysis in Real-Time Operations Optimization. *Ind. Eng. Chem. Res*., 37, 3670-3684.

Ronholm, L and Marlin, T. E. (2002). Move Moderation in Real-Time Optimization. *ESCAPE-12*, The Hague, The Netherlands.

Wolbert, D., Joulia, X., Koehret, B., and Biegler, L. T. (1994). Flowsheet Optimization and Optimal Sensitivity Analysis Using Analytical Derivatives. *Comp. Chem. Eng.*, 18(11/12), 1083-1095.

Yip, W. S. and Marlin, T. E. (2002). Multiple Data Sets for Model Updating in Real-Time Operations Optimization. *Comp. Chem. Eng.* 26(10), 1345-1362.

PREDICTIVE SCHEDULING OF A PENICILLIN BIOPROCESS PLANT

S. Lau, M.J. Willis, G.A. Montague and J. Glassey

School of Chemical Engineering & Advanced Materials,
University of Newcastle, Newcastle-upon-Tyne, NE1 7RU, UK

Abstract: In this paper a predictive strategy for the reactive scheduling of a multi-stage bioprocessing plant is outlined. In the procedure, the various batch stages of the bioprocess are dynamically re-allocated to the appropriate processing units in response to the biological variability inherent in each stage. Forecasts of the process productivity and consequent completion times for the tertiary stages of industrial penicillin fermentations are used in conjunction with a genetic algorithm to solve the scheduling problem. Initial results using data from a commercial penicillin plant demonstrate that the predictive scheduling framework could deliver increased production and, consequently, major financial benefits.

Keywords: genetic algorithms, fermentation processes, optimisation problems, scheduling algorithms, forecast

1. INTRODUCTION

The industrial scheduling problem involves assigning resources to tasks over a (fixed) period of time to fulfil production goals. Scheduling problems are often complicated by a large number of constraints. These constraints may be resource, capacity or production related. A scheduling system makes decisions dynamically to accommodate activities within the framework of available resources to ensure that tasks are completed either at a given time or at a minimum cost. Problems with scheduling occur over a wide range of industries. Examples of these are: filling and emptying of tanks in a ballast water treatment facility at a port (Dahal *et al.* 2001), clothing production (Dessouky *et al.* 1998), scheduling and rearrangements of tasks on field-programmable gate arrays (FPGA) (Middendorf *et al.* 2002), scheduling of multi-product plants (Sand *et al.* 2000) and laboratory management and scheduling of experiments to be performed on chemical workstations (Aarts *et al.* 1995). There is a vast amount of literature providing different solution approaches. Much of the literature is based on non-chemical processing applications e.g. machine shops, discrete assembly manufacture and computer system operation, but due to similarities in the problem structure, some of the literature is applicable to scheduling in the process industries (Reklaitis 1982 and Shah 1999).

Scheduling techniques can be observed from at the most basic level, manual approaches to the more sophisticated artificial intelligence strategies. An overview of these techniques is given by Morton and Pentico (1993). However a common problem that arises with scheduling is the continual need to alter previous schedules due to process problems and production changes that occur. Investigation into reactive scheduling and predictive scheduling in the manufacturing environment is discussed by Sabuncuoglu and Bayiz (2000) and Kizilisik (1999) with a special emphasis on single machine analysis. Fang *et al.* (1993) describes rescheduling in a job shop environment with the application of a genetic algorithm (GA).

The process industry covers a wide range of fields such as chemicals, pharmaceuticals, foods, paints and many others. This paper concentrates on the scheduling of operations on a batch fermentation pharmaceutical plant. There is great interest in the batch processing area as discussed by Orçun *et al.* (2001). It is known that batch processes exhibit a certain degree of variability which is caused by, for instance, changes in operator response time; fluctuations in utility availability; minor equipment malfunctions; recipe inaccuracies and changes in raw material quality (Cott and Macchietto 1989). Fermentation batches are amongst the most challenging of batch operations to schedule effectively due to the high level of inherent biological variability.

This study investigates the integration of a scheduling tool with a forecaster of production levels. At present, there is considerable variation in product concentrations (titres) with respect to processing time

in the tertiary fermentation stage (i.e. the main production stage). In particular, some batches take less time to reach the production optimum, while some are left too long and the concentration of the product starts to decline. The current process scheduling method does not take this into account and only uses nominal batch processing times derived from historical plant data. However, to optimise the production within the plant, it is necessary to predict the best time to harvest the product with respect to overall plant output and to re-schedule efficiently so to maximise production.

2. PROCESS DESCRIPTION

The plant shown in Figure 1 is used to produce the antibiotic penicillin. Each batch goes through three main stages of the fermentation process. The first two stages of the fermentation involve the growth of the organism (*Penicillium chrysogenum*). This is to allow time for it to increase its biomass and to adapt to change in volume/size and its environment (changing from growing on solid to liquid medium). The tertiary (final) fermentation stage is the production stage of the product penicillin. The choice of harvesting time for this stage is of great importance because, due to the unstable nature of penicillin, its concentration reduces after it has reached its peak in production. This has important implications for the use of any automated scheduling system. An effective scheduling mechanism must be able to take into account the variability associated with the optimum harvest time, as well as other operational factors such as contamination, maintenance, breakdowns and the like.

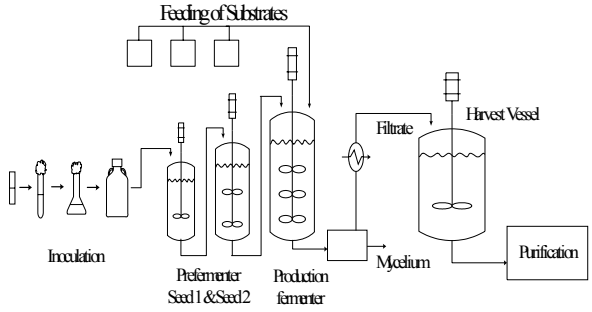


Fig 1. A simplified diagrammatic layout of a penicillin process plant.

2.1. Problem formulation

As the aim of this work is to schedule the various stages of batch operation on a bioprocessing plant, it is necessary to be able to compute the total completion time of all penicillin batches given the allocation of resources dictated by the current schedule. In addition to the nominal completion times for each of the individual stages, factors such as unit clean out times, unit setup times as well as the processing history of the plant determine the total time required to complete all batches. In the case of the predictive strategy, the predicted completion time of the each tertiary processing stage is used in place of the nominal completion time.

In this paper, a simple recursive algorithm, referred to as a recurrence relationship, with a generic structure is used as the basis for the scheduling problem solution. The generic structure means that it can be easily configured for a wide range of scheduling problems of industrial importance. Ku *et al.* (1987) developed similar relationships for jobshop problems, while Kim *et al.* (1996) suggested an algorithm for flowshop problems; however, it took the form of several *if-then* rules and therefore is not as easily configurable. In contrast, the calculation of the total completion time in this paper uses a more convenient matrix/vector formulation.

To determine the batch completion time the batch sequence is characterised by a permutation of integers 1,2,3,..,n. Let f_{nI} denote the time at which the n^{th} batch in the sequence leaves stage I. To calculate f_{nI} two conditions must be fulfilled:

1. The n^{th} batch in the sequence cannot leave a stage I unit until all the 'processing' is complete and to be on a stage I unit it must have left stage I-1
2. The n^{th} batch in the sequence can only start on stage I after one of the previously scheduled batches has finished, the unit has been cleaned, and setup is complete and cannot leave stage I until it has been processed.

Consider the following definitions:

n = total number of batches
M = total number of processing stages
P_I = total number of units available for processing a batch at stage I
$C_I(n)$ = a (1 x P_I+n-1) vector of stage I cleaning times relating the current batch to the previous 1,..,n-1 batches in the sequence, i.e.

$$C_I(n)=[c_I(1),c_I(2),..c_I(n\text{-}1)]$$

$f_I(n)$ = the stage I completion time of the n^{th} batch in the sequence
$F_I(n)$ = a (1 x P_I+n-1) vector of stage I completion times for the previous 1,..,n-1 batches in the sequence.
$r_I(n)$ = are (P_I x 1) vectors used to define the unit allocation of batch n in the sequence at stage I of the process. (if the jth element of element $r_I(n)$ = 1 then batch n is allocated to unit j at stage I and the other elements of $r_I(n)$ is equal to zero)
$t_I(n)$ = a (1 x P_I) vector defining the stage I processing time for all available units.

To calculate the completion time of batches scheduled on a M stage plant with P_I units at stage I the following recurrence relationship is used:

$$f_I(n)=t_I(n)r_I(n)+ \max(f_{I\text{-}1}(n),r_I^T(n)R_I(n)[F_I^T(n)+C_I^T(n)]) \quad (1)$$

with:

$$R_I(n)=[R_I(n\text{-}1)\text{-}r_I(n\text{-}1)\{r_I^T(n\text{-}1)R_I(n\text{-}1)\},r_I(n\text{-}1)] \quad (2)$$

The first part of the relation $f_I(n) = t_I(n)r_I+f_{I-1}(n)$ accounts for condition (1), while the second term $f_I(n) = t_I(n)r_I(n)+r_I^T(n)R_I(n)[F_I^T(n)+C_I^T(n)])$ accounts for condition (2). Operation $r_I^T(n)R_I(n)F_I^T(n)$ is used to calculate the completion time of the previous batch scheduled on the same stage I unit as the n^{th} batch in the sequence, while $r_I^T(n)R_I(n)C_I^T(n)$ determines the necessary cleaning time and setup time between two batches. Thus, $R_I(n)$ is a matrix used to indicate which of the previous *n-1* batches were last allocated to a particular unit. Note that the $R_I(n)$ is an augmented matrix comprising a modified $R_I(n-1)$ and the unit allocation vector, $r_I(n-1)$, which indicates the unit used by the previous batch. As $R_I(n)$ is used to indicate the last batch allocated to each unit, by definition $r_I(n-1)$ will be the new indicator for unit '*j*', hence the modification,

$$R_I(n-1)\text{-}r_I(n-1)\{r_I^T(n-1)R_I(n-1)\}$$

is used to remove the previous indicator. The algorithm is initialised with $R_I(0) = I$ (P_I x P_I) which will allow the calculation of the completion times for the batches already running on the plant.

2.2. Plant Information

Current plant operation uses fixed (nominal) processing times for each stage. As discussed earlier this is not ideal, as it limits the overall performance of the plant. Consideration of a number of batches of process data of actual penicillin titre (Figure 2) underpins the need to move from fixed batch times. This clearly demonstrates how each batch varies in the rate of production. For instance, with respect to batch four the peak concentration is reached at around 200hrs while for batch two the concentration of penicillin is still rising. For the purposes of this paper, such existing information is used to establish the benefits that may be obtained by stopping at the appropriate production levels rather than running for a fixed period of time.

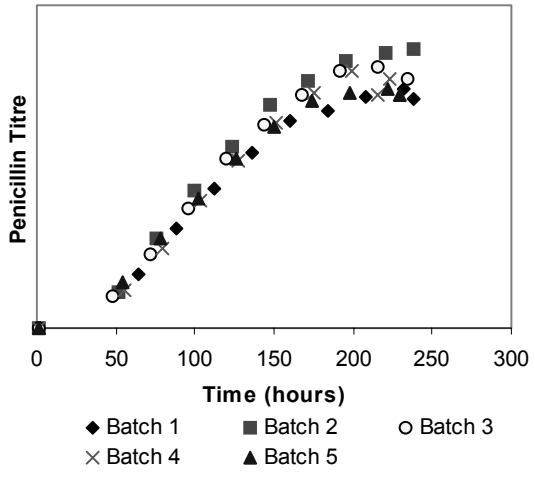


Fig. 2. Penicillin titre of five batches

A fundamental problem in the scheduling of fermentation batches is that variability leads to deviations in ideal completion time. However, it is necessary to know in advance the completion time as the previous stage must be inoculated well prior to production stage completion in order to be in an optimal state for transfer when the new production batch requires it. Thus a predictor of future penicillin titre is essential for scheduling purposes.

A further complication arises from the fact that penicillin titre is only available infrequently by off-line analysis. As a consequence the predictor is required to overcome measurement delay in addition to the horizon for scheduling requirements. One possible solution is to use estimators to determine the current penicillin titre and then use a forecaster for prediction from that point. The application of neural networks as an estimator for the penicillin titre is discussed by Yuan and Vanrolleghem (1998) and Lopes and Menezes (1998). Also, neural network models have been applied in biomass estimation and fault diagnosis for penicillin production as shown by Montague and Morris (1994). Currently estimators of current penicillin titre have not been considered, with the forecaster acting on off-line assays.

The penicillin titre trends have a distinctive shape that closely matches that of the logistic function (equation 3). The forecaster operation involves fitting the coefficients of the logistic function to penicillin titre measurements for the batch in question up to the current time and using the model to forecast future production. Forecasts are not possible in the early stage of the batch as few titre measurements are available but for scheduling purposes forecasts after around 150hrs of operation are sufficient as it is the latter period of the batch that is important.

$$y = \frac{C}{1 + Ae^{-Bx}} \tag{3}$$

The forecaster must be capable of prediction over a forty hour horizon in order to allow the necessary time for the seed batch to be completed. Figure 3 shows the forecaster applied to an example batch. It can be seen that the required forecasting capability is achieved.

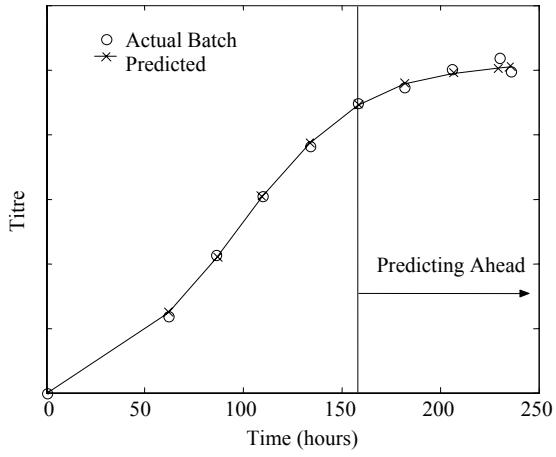


Fig. 3. Batch prediction with logistic function

The decision as to the optimum time to harvest is influenced by the forecast of the rate of penicillin production. When the rate of penicillin production falls below average rate achieved for a new batch (including the batch turn-around time) then the current batch should be terminated. However the scheduler requires the use of the forecaster to determine when this condition is likely to occur and therefore initiate a secondary stage fermentation. When the forecast is made, the best assumption is that a new batch will perform in an average manner. As the secondary stage fermentation progresses assessments of its quality may be possible and if different from average this would impact on termination time of the current tertiary stage vessel. Given the limited supply of information from the secondary stage fermentations, such modifications are currently not considered appropriate.

2.3. Solution of the scheduling problem

As noted earlier, there are many techniques available for the solution of scheduling problems. One method is the use of a genetic algorithm (GA) as this kind of algorithm has been applied successfully to many combinatorial optimisation problems (Azzaro-Pantel *et al.* 1998). The application of GA to scheduling problems has been described in the job-shop environment (Rubin and Ragatz 1995, Lee and Choi 1995 and Della Croce *et al.* 1995). Cartwright and Tuson (1994) implemented a GA to handle an industrial flowshop by optimising both chemical feed order and topology. In their study the GA provided a reliable method for finding near optimum feed order/topology combinations.

2.4. Genetic algorithms (GAs)

Genetic algorithms (GAs) are a type of heuristic optimisation method that are based on the mechanics of genetics (Holland 1975). GAs solve problems by using a process analogous to natural selection to evolve candidate solutions which are typically encoded as a population of abstract mathematical chromosomes e.g. binary, integer, or real valued string sequences. For further details see Goldberg (1989). In the following section, the type of GA that is used to schedule the batch processes is discussed.

Encoding Scheme*:* The order based GA is used, two separate strings are applied to represent batch order and unit allocation. A permutation representation is used for the batch order string, while the actual unit number is used for the unit string. Thus, the following strings are used e.g. to represent six batches and 4 units:

Batch order string	3	4	1	5	6	2
Unit string	2	4	1	3	2	1

This shows that batch 3 is processed on unit 2 first, batch 4 on unit 4 etc.

GA operators *Crossover:* The purpose of this operator is to combine information from relatively successful strings in order to produce better offspring. The classic one and two point crossover operators cannot be applied, as infeasible solutions would be generated. This appears often in strings with permutation. To avoid this a number of crossover schemes have been developed. Goldberg (1989) discusses partially matched crossover (PMX), cycle crossover (CX) and order crossover (OX) and Syswerda (1989) describes uniform crossover. In this study the PMX operator is used. Here, two cutting sites are chosen randomly (point 2 and point 5) PMX defines a matching section that is used to cross through position-by-position exchange operations. This is demonstrated below;

Parent 1	Batch1 string	3	4	1	5	6	2
	Unit1 string	2	4	1	3	2	1
Parent 2	Batch2 string	1	5	2	3	4	6
	Unit2 string	3	1	1	2	4	2

The crossover takes place with a full set of permutations within the batch order string. The unit string is then realigned with the batch order string. The offspring obtained are:

Child 1	Batch1 string	1	6	2	3	4	5
	Unit1 string	1	2	1	2	4	3
Child 2	Batch2 string	2	3	1	5	6	4
	Unit2 string	1	2	1	3	2	4

Mutation: The algorithm uses three types of mutation operator. The first is inversion where both batch order and unit string are simultaneously cut at two points and the information is reversed. The second is batch order mutation where only the batch order string is mutated and the third is unit string mutation where the unit string is mutated. The probability that a particular mutation operator is chosen is equal.

2.5. The Plant Scheduler

Figure 4 illustrates the functionality of the predictive/reactive scheduler. Given a set of process batches and unit resources, the GA optimises the schedule to produce an optimal sequence as well as determining the unit allocation for each of the batches.

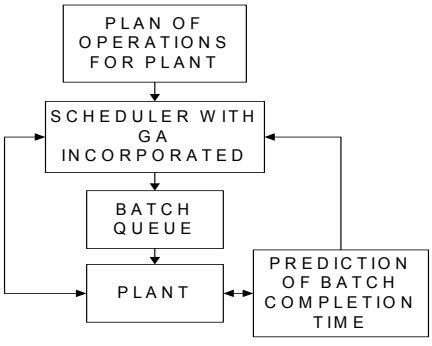


Fig. 4. The plant scheduling system

The first batches are allocated to plant resources and are processed. The remaining jobs are put into the queue (in the order defined by the schedule) and will

be processed on the plant when a processing resource becomes available. Thus, the processing time of the batches through each of the stages is monitored and their status is fed back into the scheduler. If the expected processing time of a batch changes from the unit processing time assumed initially there then presents an opportunity to re-schedule. Consequently, the GA then re-optimises the batches held in the queue to generate a new schedule that may be implemented on the plant.

2.6. Case Study

In this work, the concept of applying the knowledge of batch production status together with a forecaster for future production levels in a predictive scheduler is performed using past batch data records from the industrial plant. The past batch data acts as a 'simulation' of behaviour and can be used since in nearly all cases the actual batches were longer than appropriate. To demonstrate the overall effect of this application, 100 batches were scheduled. The batches were to be optimally sequenced through 3 stages, the first and second stage having 4 units and the final tertiary stage 18 units. The predictive scheduler is to be compared with two other schedulers; on plant scheduler and fixed GA scheduler, both of which uses fixed batch processing time. This in order will show the benefits that may be obtained by adopting a more reactive scheduling methodology.

An initial number of batches are placed onto the plant simulation. A queue is formed, as further batches are ready to be scheduled onto the plant simulation. As batch status of the main fermentation is being measured the predicted information of completion time is fed back to the scheduler. 50 runs of the GA based scheduler are used for the 100 batches. This is because different batch runs may have different completion times (as the solution may represent a local rather than global minimum) and therefore a number of runs will allow a fair comparison to be made rather than producing a one-off result.

For all runs, the GA is configured with the following parameters: crossover probability 0.8, mutation probability 0.1, population size 500, and a steady-state reproduction having a population retention of fifty individuals at each generation and each run of the GA is configured for 100 generations.

3. RESULTS

The results obtained after 50 runs are displayed by the box and whisker plots in Figure 5. The line that runs across the figure represents the time taken on plant to complete 100 batches, using the existing scheduling policy. These plots highlight the most salient features of the results obtained from each set of 50 runs. The centreline of the box is the median value of the data, whilst the box itself represents the inter quartile range of the data. From the top and bottom of each box a vertical whisker extends to the extreme values of the data. Furthermore, the notches around the median lines are constructed such that, if there is no overlap between the notches, the medians are significantly different at the 95% confidence level.

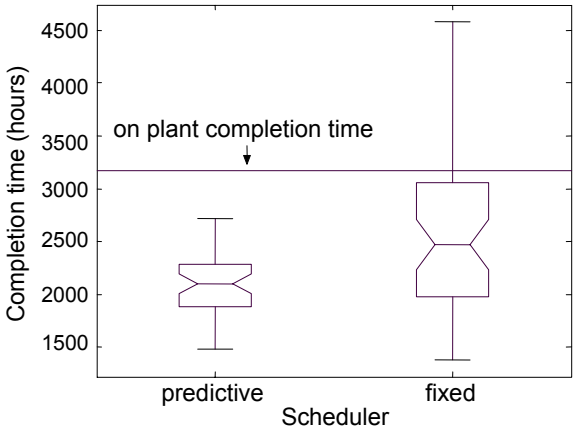


Fig. 5. Comparison of completion times between the predictive and fixed scheduler.

It can be observed from the plots produced in Figure 5 that the predictive scheduler shows a reduction in completion time for the 100 batches in comparison to the fixed scheduler. The distribution of completion times is tighter as the scheduler plans operation in a manner that takes account of DSP availability. The fixed scheduler refers to fixed operational batch length, which is sought but not achieved. Here, batch length variation is a result of schedule limitations and the inability to send the batch to DSP.
Table 1 summarises the mean completion times taken from the three schedulers (on plant, fixed GA and predictive GA scheduler).

Table 1. Mean completion times of processing 100 batches using 3 different schedulers

	On plant	Fixed	Predictive
Mean Completion Times (hrs)	3168	2536	2094

Using the predictive scheduler a saving of approximately 44 days is gained when compared to the on plant scheduler and approximately 18 days when compared to the fixed scheduler over a five month period. The days gained could allow additional batches to be processed. More importantly, as the batches were processed at their near optimum production level, clearly an improvement in the total amount of penicillin produced by the overall process is achieved.

4. CONCLUSIONS

In this paper, predictive scheduling of a bioprocess plant has been considered. The reasons behind the need for predictive scheduling have been discussed and the particular problems encountered as a result of the

biological nature of the process considered. It has been demonstrated that the use of such a predictive scheduling strategy will be beneficial. As a result the plant efficiency and productivity could potentially be significantly improved.

A key aspect of the scheduler is the forecasts of future productivity. Whilst in many batch chemical operations failure to transfer at the 'correct' time can result in reduced plant occupancy, in biological systems failure to transfer can additionally mean major productivity losses due to irreparable biological consequences. Whilst this paper attempts to address the problem of forecaster reliability, several aspects still remain to be considered. For instance a research challenge will be how to incorporate the increasing uncertainty associated with the extending forecasting horizon. Application of other models for predicting penicillin could be compared for forecasting accuracy. Subsequent work will consider how the development of an on-line forecaster to allow advance prediction with associated uncertainty will impact on the scheduler. Furthermore, work has recently identified significant variations in inoculum quality which would impact on future production performance and thus optimal completion times. Assessment of secondary stage fermentation information to update completion is an obvious improvement.

This paper has begun to address the issues associated with biological process scheduling on a complex multi-unit fermentation plant but many further advances are still required before an effective predictive scheduler suitable for the industrial environment is achieved. The scale of operation and the major financial opportunities that schedule improvement would provide a strong motivation for tackling the problems.

5. REFERENCES

Aarts, R. J., J. S. Lindsey, *et al*. (1995). Flexible Protocols Improve Parallel Experimentation Throughput. *Clinical Chemistry* **41**(7): 1004-1010.

Azzaro-Pantel, C., L. Bernal-Haro, *et al*. (1998). A Two-Stage Methodology for Short-Term Batch Plant Scheduling: Discrete-Event Simulation and Genetic Algorithm. *Computers and Chemical Engineering* **22**(10): 1461-1481.

Cartwright, H. M. and A. L. Tuson (1994). Genetic Algorithms and Flowshop Scheduling: Towards the development of a real-time process control. *AISB workshop on Evolutionary Computing.*

Cott, B. J. and S. Macchietto (1989). Minimizing the Effects of Batch Process Variablity using Online Schedule Modification. *Computers & Chemical Engineering* **13**(1/2): 105-113.

Dahal, K. P., G. M. Burt, *et al*. (2001). A Case Study of Scheduling Storage Tanks Using a Hybrid Genetic Algorithm. *IEEE Trans. Evolutionary Computation* **5**(3): 283-294.

Della Croce, F., R. Tadei, *et al.* (1995). A Genetic Algorithm for the job shop problem. *Computers & Operations Research* **22**(1): 15-24.

Dessouky, M. I., R. L. Marcellus, *et al.* (1998). Scheduling Identical Jobs on Uniform Parallel Machines with Random Processing Times. *Computers Ind Engineering* **35**(1/2): 109-112.

Fang, H.-L., P. Ross, *et al.* (1993). A Promising Genetic Algorithm Approach to Job-Shop Scheduling, Rescheduling, and Open-Shop Scheduling Problems. *Proceedings of the Fifth International Conference on Genetic Algorithms,* San Mateo, Morgan Kaufmann.

Goldberg, D. E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*., Addison Wesley.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.

Kim, M., J. H. Jung, *et al*. (1996). Intelligent Scheduling and Monitoring for Multi-product Networked Batch Processes. *Computers & Chemical Engineering* **20**(Suppl): S1149-S1154.

Kizilisik, O. B. (1999). Predictive and Reactive Scheduling, Department of Industrial Engineering, Bilkent University, TR-06533 ANKARA**:** 1-8.

Ku, H.-M., D. Rajagopalan, *et al*. (1987). Scheduling in Batch Processes. 35-45.

Lee, C. Y. and J. Y. Choi (1995). A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weights. *Computers & Operations Research* **22**(8): 857-869.

Lopes, J.P and J.C. Menezes (1998). Intelligent Systems for Penicillin Fermentation Process Modelling. *Computer Applications in Biotechnology* CAB7, 333-338.

Middendorf, M., B. Scheuermann, *et al.* (2002). An Evolutionary Approach to Dynamic Task Scheduling on FPGAs with Restricted Buffer. *Journal of Parallel and Distributed Computing* **62**(9): 1407-1420.

Montague, G. A. and J. Morris (1994). Neural-network contributions in Biotechnology. *Trends in Biotechnology* **12**: 312-324.

Morton, T. E. and D. W. Pentico (1993). *Heuristic Scheduling Systems:With Applications to Production System and Project Management*, John Wiley & Sons, INC.

Orçun, S., I. K. Altinelb, *et al*. (2001). General continuous time models for production planning and scheduling of batch processing plants: mixed integer linear program formulations and computational issues. *Computers & Chemical Engineering* **25**(2-3): 371-389.

Reklaitis, G. V. (1982). Review of Scheduling of Process Operations. *AIChE Symposium series* **78**(214): 119-132.

Rubin, P. A. and G. L. Ragatz (1995). Scheduling in a sequence dependent setup environment with genetic search. *Computers & Operations Research* **22**(1): 85-99.

Sabuncuoglu, I. and M. Bayiz (2000). Analysis of reactive scheduling problems in a job shop environment. *European Journal of Oerational Research* **126**: 567-586.

Sand, G., S. Engell, *et al.* (2000). Approximation of an Ideal Online Scheduler for a Multiproduct Batch Plant. *Computers & Chemical Engineering* **24**: 361-367.

Shah, N. (1999). Single-Multisite Planing and Scheduling:Current Status and Future Challenges. *AIChE Symposium series* **320**(94): 75-90.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*., San Mateo, California, Morgan Kaufmann.

Yuan, J. Q. and P. A. Vanrolleghem (1998). One-Step-Ahead Product Predictor for Profit Optimisation of Penicillin Fermentation. *Proceedings 7th IFAC Conference on Computer Applications in Biotechnology*, Osaka Japan.

MODELING AND OPTIMIZATION FOR HIGH-THROUGHPUT-SCREENING SYSTEMS

Eckart Mayer [1] **Jörg Raisch**

Systems and Control Theory Group, Max-Planck-Institut Dynamik komplexer technischer Systeme, 39106 Magdeburg, Germany, Fax: +49-391-6110-399
Lehrstuhl für Systemtheorie technischer Prozesse Otto-von-Guericke-Universität 39016 Magdeburg, Germany

Abstract: The problem of cyclic scheduling under the requirement of throughput maximization is considered for a special class of cyclically repeated batch processes. All batches have to follow an identical time scheme. The same resource may be visited more than once by the same batch and time window constraints may be stated by the user. It is shown that the cyclic scheduling problem can be transformed into a mixed integer linear optimization problem. The method's application to High-Throughput-Screening processes is demonstrated.

Keywords: High-Throughput-Screening, cyclic scheduling, throughput maximization, mixed integer optimization, discrete event systems

1. INTRODUCTION

Throughput maximization problems are common in many processes in chemical industry as well as in transportation or manufacturing systems, where a large number of units, e. g. batches or workpieces, have to be handled one after each other in the shortest possible time. This contribution deals with throughput maximization for a special type of cyclic systems where all units have to be handled in exactly the same time scheme. The method is applied to High-Throughput-Screening (HTS) systems. However, it is also applicable to similar cyclic processes, e. g. in tra c engineering or for iterative batch processes in chemical engineering.

High-Throughput-Screening plants are used for the analysis of large numbers of substances, for example to analyze their bene t for a speci c pharmaceutical, biological or agricultural application. Although several hundreds of substances are aggregated within one batch, i. e. on one so called microplate, a large number of batches have to pass through the plant resources, e. g. incubators, liquid handling devices, transportation devices etc., in the same speci c time scheme. The task of throughput maximization is to nd an operating sequence which allows to nish work for all microplates as fast as possible. The HTS scheduling problem di ers from other scheduling problems, e. g. in manufacturing or chemical engineering (Schilling and Pantelides, 1999; Löhl *et al.*, 1998) as it combines the following requirements:

- Some resources may be revisited several times by the same batch.
- There are no bu ers between the resources. In contrast, a batch will allocate two resources simultaneously while being transferred between the resources.

[1] Supported by CyBio AG, Jena and the German Ministry of Economics and Technology.

- All batches have to pass the system in the same time scheme.
- The time scheme may be restricted by *due dates* or *time window* constraints.

Scheduling methods exist for several xed types of HTS plants, e.g. (Murray and Anderson, 1996; Donzel *et al.*, 1997). Nevertheless, because of the large variety of screening tasks performed on HTS plants, it is very important to have exible plants with the possibility of re-arranging the machines and transportation devices in order to adapt them to the requirements of each speci c screening task. Thus, this paper presents a general method which yields the time-optimal sequence for arbitrary machine arrangements and screening tasks.

In many cases, due to the speci c nature of the substances to be screened, operating schemes have to be strictly cyclic. Thus, the method presented here will be limited to such strictly cyclic operation, where the time distance between two consecutive batches ('*cycle time*') is always constant. All resources have capacity one, i. e. each resource may be occupied by at most one batch at any one time. For such a system, the goal of throughput maximization is equivalent to the determination of the smallest possible cycle time and the corresponding batch time scheme complying with all constraints. The basic ideas for the solution of this scheduling problem have already been presented in (Mayer and Raisch, 2003). In this contribution, the method is generalized as the sequence of activities within the single batch is not xed in advance.

This paper is arranged as follows: rst, modeling of cyclic processes with respect to High-Throughput-Screening is discussed. Subsequently, the constraints for the scheduling problem are formulated. The scheduling problem is then cast into a mixed integer linear optimization problem. Finally, a speci c application example is treated.

2. MODELING OF CYCLIC PROCESSES

Cyclic operating sequences as regarded here are characterized by the fact that all operations are repeated in a constant cyclic scheme. The cycles, called batches in chemical engineering, follow upon each other with constant time o set, called *cycle time* T. In High-Throughput-Screening, one batch consists of one or a couple of microplates passing through several work steps on several resources, e. g. incubators, liquid handling devices, transport devices etc. For the screening results to be meaningful and comparable, the time scheme for all batches has to be identical. Figure 1 gives a simple example for such a time scheme for one batch (called *single-batch time scheme*). It

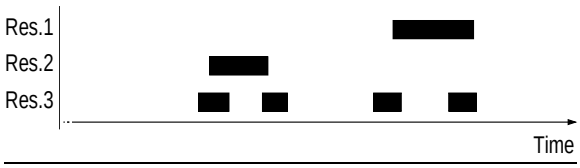


Fig. 1. Example for single-batch time scheme.

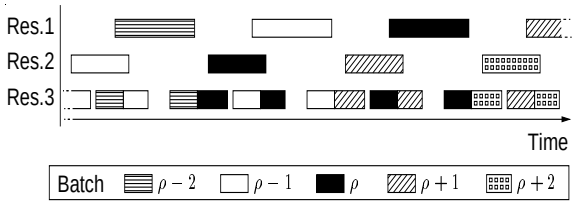


Fig. 2. Extract from cyclic schedule for Figure 1.

involves 6 activities on a total of 3 resources, pictured as a Gantt-Chart. In Figure 2, an extract of a cyclic schedule is illustrated. Di erent batches are displayed in di erent graphical patterns. Note that, due to the nature of the problem, the methods presented here do not need to account for the overall number of batches, nor do they need to identify a starting batch. Hence, batches are numbered by , $\in \mathbb{Z}$.

2.1 Modeling of Single-Batch Time Scheme

For the purpose of scheduling, the time scheme for a single batch is de ned via the time instants at which each activity starts and ends [2]. In general, it consists of i_{max} activities, each allocating one of m resources, where J_i is the resource allocated by activity i, and n_j is the number of activities on resource j during a single-batch time scheme:

o_i ... time, when activity i starts.
r_i ... time, when activity i ends, $r_i > o_i$.
J_i ... resource allocated by activity i, $J_i \in \{1 \dots m\}$.
m ... overall number of resources.
n_j ... number of activities on resource j.

For a cyclic schedule, the time instants for the -th batch are given by:

$$\begin{aligned} o_i^{(\rho)} &= o_i + \quad \cdot T \\ r_i^{(\rho)} &= r_i + \quad \cdot T\,, \qquad \in \mathbb{Z}\,,\ \ i = 1 \dots i_{max}\,, \end{aligned} \tag{1}$$

where T is the constant cycle time.

[2] This is not necessarily identical to the moment in which the microplate enters resp. leaves the resource because there may be additional pre- or post-processing.

2.2 Parameterization

If the values for the variables o_i and r_i, $i = 1 \dots i_{max}$, are all predetermined, the problem can be solved by simple algorithms in polynomial time without the need of mixed integer optimization. However, in most cases the user will not determine the entire single-batch time scheme, but will only provide some of the activity start and end times and/or a number of linear constraints on the set of possible values for the variables o_i resp. r_i. The latter is to guarantee certain sequence constraints or to cope with chemical speci cations (e. g. time windows for incubation times). The degrees of freedom that remain for the variables o_i and r_i are represented by K time variables $\theta_k \in \mathbb{R}_0^+$. The time instants o_i and r_i are then expressed as linear combinations of the θ_k:

$$\begin{aligned} o_i &= \underline{o} + \sum_{k=1}^{K} (\chi_{i,k} \cdot \theta_k) \dots \text{Start of activity} \\ r_i &= \underline{r} + \sum_{k=1}^{K} (\psi_{i,k} \cdot \theta_k) \dots \text{End of activity} . \end{aligned} \tag{2}$$

This means that the single-batch time scheme is described by xed parameters $\underline{o}$, $\underline{r}$, $\chi_{i,k}$, and $\psi_{i,k}$, $i = 1 \dots i_{max}$, and yet unknown variables θ_k, $k = 1 \dots K$. The latter could, for example, be interpreted as delays which are inserted into the sequence of activities for a microplate.

The sequence and time window constraints on the variables o_i and r_i are represented by upper bounds for the time variables θ_k,

$$\theta_k \le \theta_{k,max} , \quad k = 1 \dots K \quad , \tag{3}$$

and, if necessary, by P additional linear constraints of the form

$$\sum_{k=1}^{K} (\kappa_{p,k} \theta_k) \le \vartheta_p , \quad p = 1 \dots P \quad . \tag{4}$$

The representation of the degrees of freedom in the single-batch time scheme by use of time variables θ_k leads to a signi cantly reduced problem size, because the number of variables o_i, r_i ($2i_{max}$ variables) is reduced to K variables θ_k, and usually $K << 2i_{max}$. Based on this description of the single-batch time scheme, the scheduling problem can now be formulated.

3. THE SCHEDULING PROBLEM

The task of nding a batch time scheme for a strictly cyclic schedule which allows for the smallest possible cycle time T, thus getting maximum possible throughput, will be called scheduling problem. As described in Section 2, two subsequent sample batches enter the plant with the xed time o set T and are processed under the same basic time scheme. This can be formulated as an optimization problem.

First, some bounds for the cycle time T are formulated. Obviously, T is a strictly positive number, but a tighter bound can be deduced from the fact that if each single activity is nished as soon as possible and the busiest resource is allocated non-stop, no further speed increase is possible:

$$T \ge T_{min} = \max_j \Big(\min_{\theta_1 \dots \theta_K} \sum_{i=1}^{i_{max}} (r_i \quad o_i) \delta_{J_i j} \Big), \tag{5}$$

$$\text{where } \delta_{J_i j} = \begin{cases} 0 \text{ for } J_i \ne j \\ 1 \text{ for } J_i = j . \end{cases}$$

An *upper bound* for T can be prescribed by the user, or can be derived from the trivial case, in which no batch is started before the previous batch is completely nished, i. e.

$$T \le T_{max} = \max_{\theta_1 \dots \theta_K} (\max_i r_i \quad \min_i o_i) \quad . \tag{6}$$

Further constraints for T result from the fact that the cycle time can never be faster than the sum of allocation times (one batch) for any resource:

$$T \ge \sum_{i=1}^{i_{max}} (r_i \quad o_i) \delta_{J_i j} , \quad j = 1 \dots m , \tag{7}$$

$$\text{where } \delta_{J_i j} = \begin{cases} 0 \text{ for } J_i \ne j \\ 1 \text{ for } J_i = j . \end{cases}$$

3.1 Disjunctive Constraints

The solution to the scheduling problem has to meet the requirement that no two di erent activities are allowed to allocate the same resource simultaneously. These constraints will be called *disjunctive constraints*. Before deducing their formulation for the optimization problem, the term *nesting level* is introduced:

The nesting level $z_{(i1,i2)}$ for each combination of two activities $(i1, i2)$, $i2 > i1$, of the same resource is de ned as

$$z_{(i1,i2)} = \lceil \frac{r_{i2} \quad o_{i1}}{T} \rceil \quad 1 , \; i2 > i1 , \; J_{i1} = J_{i2} , \tag{8}$$

where $\lceil x \rceil$ denotes the ceil-function, i.e. the smallest integer number that is greater or equal to x.

The nesting level is always an integer number i. e. $z_{(i1,i2)} \in \mathbb{Z}$. For $z_{(i1,i2)} \ge 0$, the nesting level can be interpreted as follows: when considering two activities $i1$ and $i2$ of the same resource

(i. e. $J_{i1} = J_{i2}$) belonging to the same batch, the nesting level $z_{(i1,i2)}$ indicates the number of activities $i1$ belonging to other batches which take place in between. Each set of values for the variables $z_{(i1,i2)}$ represents one possible *sequence* of activities in the overall schedule.

The requirement of ruling out the overlapping of two activities can be formulated as an exclusive OR term: for two activities $(i1, i2)$ of the same resource $J_{i1} = J_{i2}$ belonging to batch 1 and 2, respectively, the following condition ensures exclusion of overlapping:

$$o_{i1}^{(\rho 1)} \geq r_{i2}^{(\rho 2)} \quad \text{XOR} \quad o_{i2}^{(\rho 2)} \geq r_{i1}^{(\rho 1)}. \qquad (9)$$

This condition has to be met for all pairs of activities on the same resource and for all pairs of batch numbers, including activities belonging to the same batch and including the same activity in di erent batches ($i1 = i2$, $_1 \neq 2$). Nevertheless, due to symmetry, it is su cient to consider only cases $i2 > i1$, as well as the special case $\{i2 = i1 = i, \ 1 \neq 2\}$. Let us consider the latter case rst. From (9), (1) and $r_i > o_i$, we get

$$\begin{aligned} r_i \quad o_i &\leq (\ 1 \quad 2) \cdot T \quad \forall(\ 1,\ 2)\,, \quad 1 > \ 2 \\ r_i \quad o_i &\leq (\ 2 \quad 1) \cdot T \quad \forall(\ 1,\ 2)\,, \quad 2 > \ 1\,, \end{aligned}$$

hence,

$$r_i \quad o_i \leq T \qquad (10)$$

for all activities i, which is already guaranteed by Equation (7).

We now investigate case 2, where (9) has to hold for all (1, 2) and for all $(i1, i2)$, $J_{i1} = J_{i2}$, $i2 > i1$. As only cyclic sequences are considered, it is su cient to ensure condition (9) for $2 = 0$ and $1 =: \ , (\ \in \mathbb{Z})$.

Equation (9) can then be reformulated using (1):

$$\begin{aligned} o_{i1} + \ T &\geq r_{i2} && (11a) \\ \text{XOR} \quad o_{i2} \qquad &\geq r_{i1} + \ T\,, \qquad \in \mathbb{Z}. && (11b) \end{aligned}$$

We now consider all possible values for , distinguishing four cases.

(1) For $= z_{(i1,i2)}$, condition (11a) will never be met. This means we need to ensure condition (11b):

$$z_{(i1,i2)} \cdot T \quad (o_{i2} \quad r_{i1}) \leq 0 \ .$$

(2) For $= z_{(i1,i2)} + 1$, condition (11b) will never be met. This means we need to ensure condition (11a):

$$\Big(\ z_{(i1,i2)} + 1 \ \Big) \cdot T \quad (r_{i2} \quad o_{i1}) \geq 0 \ .$$

(3) For $> z_{(i1,i2)} + 1$, condition (11a) is always met: from De nition (8), it immediately follows that

$$z_{(i1,i2)} + 1 \geq \frac{r_{i2} \quad o_{i1}}{T} \ .$$

Therefore

$$> \frac{r_{i2} \quad o_{i1}}{T} \ ,$$

which, in turn, implies (11a). As condition (9) is satis ed, we don't need to introduce any further conditions into the optimization problem.

(4) For $\leq z_{(i1,i2)} \quad 1$, condition (11b) is always met: from de nition (8), it follows that

$$z_{(i1,i2)} < \frac{r_{i2} \quad o_{i1}}{T} \ . \qquad (12)$$

Equation (7), ensures that $T \geq r_{i2} \quad o_{i2} + r_{i1} \quad o_{i1}$. Substituting (7) into (12) gives

$$z_{(i1,i2)} \quad 1 < \frac{o_{i2} \quad r_{i1}}{T} \ .$$

This implies

$$< \frac{o_{i2} \quad r_{i1}}{T}\,,$$

and therefore (11b). Again, condition (9) is satis ed. We don't need to introduce any further conditions into the optimization problem.

In summary, if (7) is satis ed, a necessary and su cient condition for (9) to hold for all (1, 2) and for all $(i1, i2)$, $i2 > i1$, $J_{i1} = J_{i2}$), is:

$$\begin{aligned} z_{(i1,i2)} \cdot T \quad (o_{i2} \quad r_{i1}) &\leq 0 && (13) \\ \Big(\ z_{(i1,i2)} + 1 \ \Big) \cdot T \quad (r_{i2} \quad o_{i1}) &\geq 0\,. && (14) \end{aligned}$$

Equations (13) and (14), together with $r_{i1} > o_{i1}$ and $r_{i2} > o_{i2}$, at the same time ensure that de nition (8) is met.

3.2 MILP Formulation

We can now formulate our scheduling problem as an optimization problem.

In order to simplify notation, each possible pair of values for indices $(i1, i2)$, $i2 > i1$, $J_{i1} = J_{i2}$, is mapped to one value for a single index ι

$$\iota = 1 \ldots \iota_{max}\ , \quad \iota_{max} = \sum_{j=1}^{m} \frac{n_j(n_j \quad 1)}{2} \ . \qquad (15)$$

Hence, each value for ι signi es a pair of activities using the same resource.

The following abbreviations are introduced:

$$v_{\iota,k} = \chi_{i2,k} \quad \psi_{i1,k} \tag{16}$$

$$w_{\iota,k} = \psi_{i2,k} \quad \chi_{i1,k} \tag{17}$$

$$v_{\iota,0} = \quad \quad \tag{18}$$

$$w_{\iota,0} = \quad \quad . \tag{19}$$

Substituting (2) into (7) results in

$$b_{j,0} + \sum_{k=1}^{K} (b_{j,k} \cdot \theta_k) \quad T \leq 0 \ , \ \ j = 1 \ldots m \ , \tag{20}$$

where

$$b_{j,0} = \sum_{i=1}^{i_{max}} (\quad) \delta_{J_i j}$$

$$b_{j,k} = \sum_{i=1}^{i_{max}} \sum_{k=1}^{K} (\psi_{i,k} \quad \chi_{i,k}) \delta_{J_i j}$$

$$\delta_{J_i j} = \begin{cases} 0 \text{ for } J_i \neq j \\ 1 \text{ for } J_i = j \ . \end{cases}$$

In order to formulate the scheduling problem as an optimization problem, we have to take the cycle time T as the objective function to be minimized under the constraints given by equations (13) and (14) as well as (3), (4), (5), (6) and (20). The search space for the optimization problem is de ned by the following variables:

- cycle time $T \in \mathbb{R}^+$,
- time variables $\theta_k \in \mathbb{R}_0^+$,
- nesting levels $z_\iota \in \mathbb{Z}$.

Hence, using the abbreviations (16) to (19) and (2), the HTS scheduling problem can be written as the following optimization problem:

Min T over ($T \in \mathbb{R}^+$, $\theta_k \in \mathbb{R}_0^+$, $z_\iota \in \mathbb{Z}$) (NP1)
s. th.

$$z_\iota \cdot T \quad v_{\iota,0} \quad \sum_{k=1}^{K} (\ v_{\iota,k} \cdot \theta_k) \leq 0 \tag{NP2}$$

$$\Big(\ z_\iota + 1 \ \Big) \cdot T \quad w_{\iota,0} \quad \sum_{k=1}^{K} (w_{\iota,k} \cdot \theta_k) \geq 0 \tag{NP3}$$

$$\text{for } \iota = 1 \ldots \iota_{max}$$

$$\theta_k \leq \theta_{k,max} \ , \ k = 1 \ldots K \tag{NP4}$$

$$\sum_{k=1}^{K} (\kappa_{p,k} \cdot \theta_k) \leq \vartheta_p \ , \ p = 1 \ldots P \tag{NP5}$$

$$T_{min} \leq T \leq T_{max} \tag{NP6}$$

$$b_{j,0} + \sum_{k=1}^{K} (b_{j,k} \cdot \theta_k) \quad T \leq 0 \ , \ j = 1 \ldots m \tag{NP7}$$

Clearly, (NP1) to (NP7) constitutes a mixed integer nonlinear program (MINLP), i. e. a nonlinear optimization problem consisting of real and integer variables. There exist several solvers for MINLP optimization problems with di erent advantages and disadvantages. However, this task is very complex and can result in long computing times, even for small systems. Fortunately, it is possible to transform the MINLP into a linear formulation using

$$\bar{T} := \frac{1}{T} \ , \ \ \bar{\theta}_k := \frac{\theta_k}{T} \ , \ \ k = 1 \ldots K \ . \tag{21}$$

This leads to the following mixed integer linear program (MILP):

Max $\bar{T}$ over ($\bar{T} \in \mathbb{R}^+$, $\bar{\theta}_k \in \mathbb{R}_0^+$, $z_\iota \in \mathbb{Z}$) (LP1)
s. th.

$$z_\iota \quad v_{\iota,0} \cdot \bar{T} \quad \sum_{k=1}^{K} (\ v_{\iota,k} \cdot \bar{\theta}_k) \leq 0 \tag{LP2}$$

$$z_\iota + 1 \quad w_{\iota,0} \cdot \bar{T} \quad \sum_{k=1}^{K} (w_{\iota,k} \cdot \bar{\theta}_k) \geq 0 \tag{LP3}$$

$$\text{for } \iota = 1 \ldots \iota_{max}$$

$$\bar{\theta}_k \quad \theta_{k,max} \, \bar{T} \leq 0 \ , \ k = 1 \ldots K \tag{LP4}$$

$$\sum_{k=1}^{K} (\kappa_{p,k} \cdot \bar{\theta}_k) \quad \vartheta_p \, \bar{T} \leq 0 \ , \ p = 1 \ldots P \tag{LP5}$$

$$\frac{1}{T_{max}} \leq \bar{T} \leq \frac{1}{T_{min}} \tag{LP6}$$

$$b_{j,0} \cdot \bar{T} + \sum_{k=1}^{K} (b_{j,k} \cdot \bar{\theta}_k) \quad 1 \leq 0 \ , \ j = 1 \ldots m \tag{LP7}$$

Such an optimization problem can be solved [3] by advanced branch-and-bound techniques, for example by using the CPLEX library (http://www.ilog.com/products/cplex).

3.3 Adding Bounds for Integer Variables

For assays with a large number of activities, the optimization problem may become rather complex [4] which can result in very long computation times. Therefore it can be helpful to add additional bounds for the integer variables z_ι to the problem (LP1) to (LP7). This can be done as follows:

$$z_{\iota,min} \leq z_\iota \leq z_{\iota,max} \ , \tag{22}$$

where [5]

---

[3] Attention has to be paid to numerical aspects during optimization runs due to the fact that now the reciprocal of the original objective function is used.

[4] For n_j activities on a resource j, the number of pairs is $\frac{n_j(n_j-1)}{2}$.

[5] $\lfloor x \rfloor$ denotes the floor-function, i.e. the largest integer number that is less or equal to x.

$$z_{\iota,min} := \begin{cases} \lceil \frac{\underline{W}_\iota}{T_{min}} \rceil & 1 \text{ for } \underline{W}_\iota < 0 \\ \lceil \frac{\underline{W}_\iota}{T_{max}} \rceil & 1 \text{ for } \underline{W}_\iota \geq 0 \end{cases}$$

$$z_{\iota,max} := \begin{cases} \lfloor \frac{\bar{V}_\iota}{T_{max}} \rfloor & \text{for } \bar{V}_\iota \leq 0 \\ \lfloor \frac{\bar{V}_\iota}{T_{min}} \rfloor & \text{for } \bar{V}_\iota > 0 \end{cases}$$

$$\underline{W}_\iota = \min_{\theta_1 \ldots \theta_K} \left(w_{\iota,0} + \sum_{k=1}^{K} w_{\iota,k} \cdot \theta_k \right)$$

$$\bar{V}_\iota = \max_{\theta_1 \ldots \theta_K} \left(v_{\iota,0} + \sum_{k=1}^{K} v_{\iota,k} \cdot \theta_k \right) .$$

Note that introducing these bounds only removes parts of the search space where at least one of conditions (LP1) to (LP7) is not satis ed, i. e. the feasible region of the optimization problem is not reduced.

4. APPLICATION

The proposed method has been applied to a number of HTS tasks (assays) in the pharmaceutical industry. An example for such an assay is given in the following: it consists of $m = 12$ resources with

$$\begin{aligned} n_j &= 8 \,, \quad j = 1, 12 \\ n_j &= 5 \,, \quad j = 5, 11 \\ n_j &= 2 \,, \quad j = 2 \ldots 4 \\ n_j &= 1 \,, \quad j = 6 \ldots 10 \end{aligned}$$

i. e. there are eight activities on the rst resource, two activities on the second resource etc. The total number of pairs of activities on the same resource, i. e. the number of integer variables z_ι, is $\iota_{max} = 79$ (see Equation (15)). For this example the number of real variables θ_k is $K = 30$. This number follows from the time window constraints speci ed by the user. Empiric approaches (e. g. shifting of activities or insertion of delays in order to nd solutions with small cycle times) will normally only provide suboptimal results for such problems. The mixed integer linear program for this example consists of 30+1 real valued variables and 79 integer variables as well as 204 linear inequality constraints. A globally optimal solution has been found using GAMS/CPLEX (http://www.gams.com) requiring a calculation time of only a few seconds.

In general, it is not possible to provide guaranteed computation time bounds for solving mixed integer optimization problems, since slight changes in the problem structure can have signi cant impact on overall computing times of solver algorithms. Nevertheless, for all HTS scheduling problems we considered, the computing times have found to be highly acceptable for the user.

5. CONCLUSION

The problem of nding a time optimal schedule for cyclically repeated batch processes with revisited resources and time window constraints has been treated. It has been shown that this scheduling problem can be modeled as a mixed integer linear optimization program (MILP). Problem instances (assays) for real High-Throughput-Screening (HTS) plants result in optimization programs of reasonable size and structure for which a globally optimal solution can be found within short calculation time.

With respect to HTS, two aspects have not been treated in this paper. The rst is how to treat multiple assays or resources with a capacity greater than one. The second is the description of methods for casting the user de ned time window constraints into the linear representation (2) using time variables θ_k such that a minimum number of variables is needed.

ACKNOWLEDGMENTS

The authors gratefully acknowledge funding by CyBio AG, Jena, Germany and the German Ministry of Economics and Labour via the PRO INNO project 'Just-In-Time Optimierung verteilter Prozessabläufe im HTS-Labor'.

REFERENCES

Donzel, A., J. Carmona and L.A. Corkan (1997). Perspectives on scheduling. In: *High throughput screening* (J.P. Devlin, Ed.). Chap. 33, pp. 525–544. Marcel Dekker Inc. New York.

Löhl, T., C. Schulz and S. Engell (1998). Sequencing of batch operations for a highly coupled production process: Genetic algorithms versus mathematical programming. *Computers and Chemical Engineering* **22, Suppl.**, 579–585.

Mayer, E. and J. Raisch (2003). Time-optimal scheduling for high throughput screening processes using cyclic discrete event models. In: *Proc. 4th Mathmod, Vienna, 2003.* pp. 1349–1356.

Murray, C. and C. Anderson (1996). Scheduling software for high-throughput screening. *Laboratory Robotics & Automation* **8**(5), 295–305.

Schilling, G. and C.C. Pantelides (1999). Optimal periodic scheduling of multipurpose plants. *Computers and Chemical Engineering* **23**, 635–655.

VARIANCE-CONSTRAINED FILTERING FOR UNCERTAIN STOCHASTIC SYSTEMS WITH MISSING MEASUREMENTS

Zidong Wang [*,1] **Daniel W. C. Ho** [**]

[*] *Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, UB8 3PH, U.K.*
[**] *Department of Mathematics, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong.*

Abstract: In this paper, we consider a new filtering problem for linear uncertain discrete-time stochastic systems with missing measurements. The parameter uncertainties are allowed to be norm-bounded and enter into the state matrix. The system measurements may be unavailable (i.e., missing data) at any sample time, and the probability of the occurrence of missing data is assumed to be known. The purpose of this problem is to design a linear filter such that, for all admissible parameter uncertainties and all possible incomplete observations, the error state of the filtering process is mean square bounded, and the steady-state variance of the estimation error of each state is not more than the individual prescribed upper bound. It is shown that, the addressed filtering problem can effectively be solved in terms of the solutions of a couple of algebraic Riccati-like inequalities or linear matrix inequalities. The explicit expression of the desired robust filters is parameterized, and an illustrative numerical example is provided to demonstrate the usefulness and flexibility of the proposed design approach.

Keywords: Kalman filtering; robust filtering; incomplete observation; missing signal; linear matrix inequality.

1. INTRODUCTION

The well-known Kalman filtering is one of the most successful H_2 filtering approaches widely used in various fields of signal processing and control. However, it has now been recognized that the standard Kalman filtering algorithm will generally not guarantee satisfactory performance when there exist parameter uncertainties in the system model. To improve the robustness, in recent years, many alternative design methods have been developed, among them we just mention the H_∞ filtering and robust filtering approaches, see for example Fu, *et al.*, 2001, Palhares, *et al.*, 2001, Shaked, *et al.*, 2001, and references therein.

In practical engineering, however, it is often the case that, for a class of filtering problems such as the tracking of a maneuvering target, the performance objectives are naturally described as the upper bounds on the error variances of estimation, see *e.g.* Skelton and Iwasaki (1993) and Yaz and Skelton (1991). Unfortunately, it is usually difficult to utilize traditional methods to deal with this class of *constrained variance* filtering problems. For instance, the theory of weighted least-squares estimation minimizes a weighted scalar sum of the error variances of the state estimation, but minimizing a scalar sum does not ensure

[1] Partially supported by City University of Hong Kong (CityU SRG No. 7001146), the EPSRC of U.K., and the Alexander von Humboldt Foundation of Germany. E-mail: Zidong.Wang@brunel.ac.uk.

that the multiple variance requirements will be satisfied (Stengel, 1986). Motivated by this fact, a novel filtering method, namely, error covariance assignment (ECA) theory (see *e.g.* NaNacara and Yaz, 1997 and Yaz and Skelton, 1991), was developed to provide a closed form solution for directly assigning the specified steady-state estimation error covariance. Subsequently, the idea of ECA theory has been applied in investigating the so-called variance-constrained filtering problems for parameter uncertain systems (Wang and Huang, 2000), sampled-data systems (Wang, *et al.* 2001), and bilinear systems (Wang and Qiao, 2001), where a prespecified upper bound is placed onto the steady-state estimation error variance.

So far, in the literature mentioned above, it is assumed that the measurements always contain the signal. However, in practical applications such as target tracking, there may be a nonzero probability that any observation consists of noise alone if the target is absent, *i.e.*, the measurements are not consecutive but contain missing observations. The missing observations are caused by a variety of reasons, *e.g.*, the high maneuverability of the tracked target, a certain failure in the measurement, intermittent sensor failures, accidental loss of some collected data, or some of the data may be jammed or coming from a high noise environment, *etc.*, see Rosen and Porat (1989).

Basically, the standard definition of covariance in the data statistical analysis does not directly apply if some of the measurements are unavailable. Thus, the popular robust and/or H_∞ filtering approaches, which are dependent on the system output covariance, do not suit the case when there are missing measurements. For filtering problem, only a very limited number of filter design methods for system output signals with missing measurements have been developed. In Kassel and Baxa (1988), the effect of missing data on the steady-state performance of a tracking filter was shown to be crucial. Chen (1990) proposed a suboptimal Kalman filtering method to cope with the case of measurement data missing. A measurement model with a binary multiplicative noise was employed in NaNacara and Yaz (1997) to study the filter design problem with error covariance assignment. Some more relevant references can also be found in Chow and Birkemeier (1990) and NaNacara and Yaz (1997). Up to now, to the best of the authors' knowledge, the issue of *variance-constrained* filtering on *parameter uncertain* systems with *missing measurements* has not been fully investigated and remains to be important and challenging.

In this paper, we are concerned with the variance-constrained filtering problem for uncertain discrete-time stochastic systems with probabilistic missing measurements. We aim at designing a linear filter such that, for all admissible parameter uncertainties and all possible incomplete observations, 1) the error state of the filtering process is mean square bounded; and 2) the steady-state variance of the estimation error of each state is not more than the individual prescribed upper bound. It is shown that, the solution to the addressed filtering problem is related to a couple of algebraic Riccati-like inequalities or linear matrix inequalities. The explicit expression of the desired robust filters is derived, and a numerical example is offered to illustrate the usefulness of the proposed design approach.

Notation. The notations in this paper are quite standard. $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ denote, respectively, the n dimensional Euclidean space and the set of all $n\times m$ real matrices. The superscript "T" denotes the transpose and the notation $X \geq Y$ (respectively, $X > Y$) where X and Y are symmetric matrices, means that $X - Y$ is positive semi-definite (respectively, positive definite). I is the identity matrix with compatible dimension. Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, P)$ be a complete probability space with a filtration $\{\mathcal{F}_t\}_{t\geq 0}$ satisfying the usual conditions (*i.e.*, the filtration contains all P-null sets and is right continuous). $\mathcal{E}\{\cdot\}$ stands for the mathematical expectation operator with respect to the given probability measure P. Prob$\{\cdot\}$ means the occurrence probability of the event "$\cdot$".

2. PROBLEM FORMULATION AND ASSUMPTIONS

Consider the following linear uncertain discrete-time stochastic system

$$x(k+1) = (A + \Delta A)x(k) + w(k), \qquad (1)$$

and the measurement equation

$$y(k) = \gamma(k)Cx(k) + v(k) \qquad (2)$$

where $x \in \mathbb{R}^n$ is a state vector, $y \in \mathbb{R}^p$ is a measured output vector, and A and C are known constant matrices. $w(k) \in \mathbb{R}^n$ and $v(k) \in \mathbb{R}^p$ are mutually uncorrelated zero mean Gaussian white noise sequences with respective covariances $W > 0$ and $V > 0$. The initial state $x(0)$ has the mean $\bar{x}(0)$ and covariance $P(0)$, and is uncorrelated with both $w(k)$ and $v(k)$. ΔA is a real-valued perturbation matrix being of the following form

$$\Delta A = MFN, \quad FF^T \leq I \qquad (3)$$

and M and N are known constant matrices of appropriate dimensions which specify how the elements of the nominal matrix A are affected by the uncertain parameters in F. The uncertainties in ΔA are said to be admissible if (3) holds. The stochastic variable $\gamma(k) \in \mathbb{R}$ is a Bernoulli distributed white sequence taking values on 0 and 1 with

$$\text{Prob}\{\gamma(k)=1\}=\mathcal{E}\{\gamma(k)\}:=\bar{\gamma} \tag{4}$$

where $\bar{\gamma}$ is a known positive constant, and $\gamma(k)\in\mathbb{R}$ is assumed to be independent of $w(k)$, $v(k)$, and $x(0)$. Therefore, we have

$$\text{Prob}\{\gamma(k)=0\}=1-\bar{\gamma} \tag{5}$$

$$\sigma_\gamma^2:=\mathcal{E}\{(\gamma(k)-\bar{\gamma})^2\}=(1-\bar{\gamma})\bar{\gamma} \tag{6}$$

Remark 1. The system measurement mode (2) has subsequently been used in many papers (see *e.g.* NaNacara and Yaz, 1997) to account for the probabilistic measurement missing.

Assumption 1. The matrix A is nonsingular and Schur stable (*i.e.*, all eigenvalues of A are located within the unit circle in the complex plane).

Introducing now a new stochastic sequence

$$\tilde{\gamma}(k):=\gamma(k)-\bar{\gamma}, \tag{7}$$

we can see that $\tilde{\gamma}(k)$ is a scalar zero mean stochastic sequence with variance

$$\sigma_{\tilde{\gamma}}^2=(1-\bar{\gamma})\bar{\gamma}. \tag{8}$$

The linear full-order filter considered in this paper is of the following structure

$$\hat{x}(k+1)=G\hat{x}(k)+K(y(k)-\bar{\gamma}C\hat{x}(k)) \tag{9}$$

where $\hat{x}(k)$ stands for the state estimate, and G and K are the filter parameters to be scheduled.

The steady-state estimation error covariance is defined by

$$P:=\lim_{k\to\infty}P(k):=\lim_{k\to\infty}\mathcal{E}[e(k)e^T(k)], \tag{10}$$

where $e(k)=x(k)-\hat{x}(k)$.

From (1)-(2), (7) and (9), we have $y(k)-\bar{\gamma}C\hat{x}(k)=\tilde{\gamma}(k)Cx(k)+\bar{\gamma}Ce(k)+v(k)$, and subsequently

$$\begin{aligned}e(k+1)=&(A+\Delta A-G-\tilde{\gamma}(k)KC)x(k)\\&+(G-\bar{\gamma}KC)e(k)+w(k)-Kv(k).\end{aligned} \tag{11}$$

Define $x_f(k):=\left[\,x^T(k)\;\;e^T(k)\,\right]^T$, and

$$A_f:=\begin{bmatrix}A & 0\\ A-G-\tilde{\gamma}(k)KC & G-\bar{\gamma}KC\end{bmatrix}, \tag{12}$$

$$A_n:=\begin{bmatrix}A & 0\\ A-G & G-\bar{\gamma}KC\end{bmatrix},\quad J:=\begin{bmatrix}0 & 0\\ \sigma_{\tilde{\gamma}}KC & 0\end{bmatrix} \tag{13}$$

$$M_f:=\begin{bmatrix}M\\ M\end{bmatrix},\quad N_f:=\begin{bmatrix}N & 0\end{bmatrix},\quad \Delta A_f=M_fFN_f, \tag{14}$$

$$W_f:=B_fB_f^T:=\begin{bmatrix}W & W\\ W & W+KVK^T\end{bmatrix}, \tag{15}$$

$$X(k):=\mathcal{E}[x_f(k)x_f^T(k)]:=\begin{bmatrix}X_{xx}(k) & X_{xe}(k)\\ X_{xe}^T(k) & X_{ee}(k)\end{bmatrix}. \tag{16}$$

Considering (1) and (11), we obtain the following augmented system

$$x_f(k+1)=(A_f+\Delta A_f)x_f(k)+B_fw_f(k), \tag{17}$$

where $w_f(k)$ denotes a zero mean Gaussian white noise sequence with unity intensity $I>0$.

Remark 2. It is mentionable that there is a stochastic variable $\tilde{\gamma}(k)$ involved in A_f, which reflects the characteristic of the missing measurement for the addressed filtering problem, and the augmented system (17) is therefore essentially a stochastic parameter system. Note that robust filtering problem for stochastic parameter systems has not gain much attention in the literature.

Using the statistics of the noises $w(k)$, $v(k)$ and, in particular, $\tilde{\gamma}(k)$, the state covariance $X(k)$ defined in (16) is found to satisfy

$$\begin{aligned}X(k+1)=&(A_n+\Delta A_f)X(k)(A_n+\Delta A_f)^T\\&+JX(k)J^T+W_f\end{aligned} \tag{18}$$

We know from Agniel and Jury (1971) and DeKoning (1984) that, if the state of the system (17) is mean square bounded, the steady-state covariance X of the system (17) defined by

$$X:=\lim_{k\to\infty}X(k) \tag{19}$$

exists and satisfies the following discrete-time modified Lyapunov equation

$$\begin{aligned}X=&(A_n+\Delta A_f)X(A_n+\Delta A_f)^T\\&+JXJ^T+W_f.\end{aligned} \tag{20}$$

Remark 3. It follows from Agniel and Jury (1971) and DeKoning (1984) that, there exists a unique symmetric positive semi-definite solution to (20) if and only if

$$\rho\big\{(A_n+\Delta A_f)\otimes(A_n+\Delta A_f)+J\otimes J\big\}<1 \tag{21}$$

where ρ is the spectral radius and $\otimes$ is the Kronecker product. Furthermore, we also know from Agniel and Jury (1971) and DeKoning (1984) that the condition (21) is equivalent to the mean square boundedness of the state of the system (17). Hence, we conclude here that, if there exists a positive definite solution to the equation (20), then (21) holds, and the convergence of $X(k)$ in (16) will be guaranteed to a constant value X.

The purpose of this paper is to design the filter parameters, G and K, such that for all admissible perturbations ΔA, 1) the state of the augmented system (17) is mean square bounded, *i.e.*, (21) holds; and 2) the steady-state error covariance X_{ee} satisfies

$$[X_{ee}]_{ii}\le\alpha_i^2,\quad i=1,2,\cdots,n. \tag{22}$$

where $[X_{ee}]_{ii}$ means the steady-state variance of the ith error state, and α_i^2 $(i = 1, 2, \cdots, n)$ denotes the prespecified steady-state error estimation variance constraint on the ith state.

In the next section, we will first characterize an upper bound on the steady-state error covariance X satisfying (20) in terms of some free parameters, and let this upper bound meet the prespecified variance constraints, and then we will parameterize all desired filter gains with which the resulting steady-state error covariance is not more than the obtained upper bound.

3. MAIN RESULTS AND PROOFS

Lemma 1. Let a positive scalar $\varepsilon > 0$ and a positive definite matrix $Q_f > 0$ be such that $N_f Q_f N_f^T < \varepsilon I$, and $\Delta A_f = M_f F N_f$ with $FF^T \leq I$. Then

$$\begin{aligned}&(A_n + \Delta A_f) Q_f (A_n + \Delta A_f)^T \\ &\leq A_n (Q_f^{-1} - \varepsilon^{-1} N_f^T N_f)^{-1} A_n^T + \varepsilon M_f M_f^T \end{aligned} \quad (23)$$

holds for all admissible perturbations ΔA.

Lemma 2. (Wang and Huang, 2000) For a given negative definite matrix $\Pi < 0$ ($\Pi \in \mathbb{R}^{n\times n}$), there always exists a matrix $L \in \mathbb{R}^{n\times p}$ $(p \leq n)$ such that $\Pi + LL^T < 0$.

Lemma 3. (Matrix Inverse Lemma) Let A, B, C and D be given matrices of appropriate dimension with A, D, and $D^{-1} + CA^{-1}B$ being invertible, then $(A + BDC)^{-1} = A^{-1} - A^{-1}B(D^{-1} + CA^{-1}B)^{-1}CA^{-1}$.

Lemma 4. (Schur complement) Given constant matrices Ω_1, Ω_2, Ω_3 where $\Omega_1 = \Omega_1^T$ and $0 < \Omega_2 = \Omega_2^T$, then $\Omega_1 + \Omega_3^T \Omega_2^{-1} \Omega_3 < 0$ if and only if

$$\begin{bmatrix} \Omega_1 & \Omega_3^T \\ \Omega_3 & -\Omega_2 \end{bmatrix} < 0, \quad \text{or} \quad \begin{bmatrix} -\Omega_2 & \Omega_3 \\ \Omega_3^T & \Omega_1 \end{bmatrix} < 0.$$

For presentation convenience, we denote:

$$\begin{aligned}\Phi &:= (A - G)(P_1^{-1} - \varepsilon^{-1} N^T N)^{-1} (A - G)^T \\ &\quad + \varepsilon M M^T + W, \quad (24)\\ R &:= (\bar{\gamma}^2 + \sigma_{\tilde{\gamma}}^2) C P_2 C^T + V, \quad (25)\\ \Pi &:= \Phi + G P_2 G^T - P_2 \\ &\quad - \bar{\gamma}^2 G P_2 C^T R^{-1} C P_2 G^T, \quad (26)\end{aligned}$$

where $\bar{\gamma}$ and $\sigma_{\tilde{\gamma}}$ are defined in (4) and (8), respectively.

Theorem 1. Assume that there exists a positive scalar $\varepsilon > 0$ such that the following two quadratic matrix inequalities

$$\begin{aligned}&A P_1 A^T - P_1 + A P_1 N^T (\varepsilon I - N P_1 N^T)^{-1} N P_1 A^T \\ &+ \varepsilon M M^T + W < 0 \quad (27)\\ \Pi &= \Phi + G P_2 G^T - P_2 \\ &- \bar{\gamma}^2 G P_2 C^T R^{-1} C P_2 G^T < 0 \quad (28)\end{aligned}$$

respectively have positive definite solutions $P_1 > 0$ ($N P_1 N^T \leq \varepsilon I$) and $P_2 > 0$, where

$$\begin{aligned}G = A &+ (\varepsilon M M^T + W)(A^{-1})^T \\ &\cdot (P_1^{-1} - \varepsilon^{-1} N^T N). \end{aligned} \quad (29)$$

Moreover, let $L \in \mathbb{R}^{n\times p}$ $(p \leq n)$ be an arbitrary matrix satisfying $\Pi + LL^T < 0$ (see Lemma 2), and $U \in \mathbb{R}^{p\times p}$ be an arbitrary orthogonal matrix (*i.e.*, $UU^T = I$). Then, the filter (9) with the parameters determined by (29) and

$$K = \bar{\gamma} G P_2 C^T R^{-1} + L U R^{-1/2}, \quad (30)$$

will be such that, for all admissible perturbations ΔA, 1) the state of the augmented system (17) is mean square bounded; 2) the steady-state error covariance X_{ee} meets $X_{ee} < P_2$.

Proof: Define $P_f := \mathrm{diag}(P_1, P_2)$. Then, it follows directly from Lemma 1 and the definitions (24)-(26) that

$$\begin{aligned}&(A_n + \Delta A_f) P_f (A_n + \Delta A_f)^T - P_f + J P_f J^T \\ &+ W_f \leq A_n (P_f^{-1} - \varepsilon^{-1} N_f^T N_f)^{-1} A_n^T + \varepsilon M_f M_f^T \\ &- P_f + J P_f J^T + W_f := \Psi := \begin{bmatrix} \Psi_{11} & \Psi_{12} \\ \Psi_{12}^T & \Psi_{22} \end{bmatrix} \end{aligned} \quad (31)$$

where

$$\begin{aligned}\Psi_{11} &= A(P_1^{-1} - \varepsilon^{-1} N^T N)^{-1} A^T - P_1 \\ &\quad + \varepsilon M M^T + W, \quad (32)\\ \Psi_{12} &= A(P_1^{-1} - \varepsilon^{-1} N^T N)^{-1} (A - G)^T \\ &\quad + \varepsilon M M^T + W, \quad (33)\\ \Psi_{22} &= (A - G)(P_1^{-1} - \varepsilon^{-1} N^T N)^{-1} (A - G)^T \\ &\quad + (G - \bar{\gamma} K C) P_2 (G - \bar{\gamma} K C)^T \\ &\quad + \varepsilon M M^T - P_2 + \sigma_{\tilde{\gamma}}^2 K C P_2 C^T K^T \\ &\quad + W + K V K^T. \quad (34)\end{aligned}$$

It follows immediately from Lemma 3 that

$$\begin{aligned}&(P_1^{-1} - \varepsilon^{-1} N^T N)^{-1} \\ &= P_1 + P_1 N^T (\varepsilon I - N P_1 N^T)^{-1} N P_1\end{aligned}$$

and therefore the inequality (27) implies that $\Psi_{11} < 0$. Moreover, substituting the expression of G in (29) into (33) leads to $\Psi_{12} = 0$ easily.

Next, we shall consider Ψ_{22}. By using the definitions (24)-(26), we can rearrange (34) as follows

$$
\begin{aligned}
\Psi_{22} &= \Phi + (G-\bar{\gamma}KC)P_2(G-\bar{\gamma}KC)^T - P_2 \\
&\quad + \sigma_{\tilde{\gamma}}^2 KCP_2C^TK^T + KVK^T \\
&= \Phi + GP_2G^T - P_2 + K[(\bar{\gamma}^2+\sigma_{\tilde{\gamma}}^2)CP_2C^T \\
&\quad + V]K^T - \bar{\gamma}GP_2C^TK^T - \bar{\gamma}KCP_2G^T \\
&= \Phi + GP_2G^T - P_2 - \bar{\gamma}^2GP_2C^TR^{-1}CP_2G^T \\
&\quad + (KR^{1/2} - \bar{\gamma}GP_2C^TR^{-1/2}) \\
&\quad \cdot (KR^{1/2} - \bar{\gamma}GP_2C^TR^{-1/2})^T \\
&= \Pi + (KR^{1/2} - \bar{\gamma}GP_2C^TR^{-1/2}) \\
&\quad \cdot (KR^{1/2} - \bar{\gamma}GP_2C^TR^{-1/2})^T. \qquad (35)
\end{aligned}
$$

Noticing the expression of $K = \bar{\gamma}GP_2C^TR^{-1} + LUR^{-1/2}$ in (30) and the fact that $UU^T = I$, we have

$$
\begin{aligned}
&(KR^{1/2} - \bar{\gamma}GP_2C^TR^{-1/2}) \\
&\cdot (KR^{1/2} - \bar{\gamma}GP_2C^TR^{-1/2})^T = LL^T.
\end{aligned}
$$

Thus, it follows from (35), the definition of the matrix L ($L \in \mathbb{R}^{n\times p}$) and the inequality (28) that $\Psi_{22} = \Pi + LL^T < 0$.

To this end, we can conclude that $\Psi < 0$. Therefore, it follows from (31) that

$$
\begin{aligned}
&(A_n + \Delta A_f)P_f(A_n + \Delta A_f)^T - P_f + JP_fJ^T \\
&\qquad \le -W_f + \Psi < 0 \qquad (36)
\end{aligned}
$$

which leads to (21). As discussed earlier Remark 3, we know that the state of the augmented system (17) is mean square bounded, and there exists a symmetric positive semi-definite solution to (20). The first claim of this theorem is then proved.

Furthermore, subtract (20) from (36) to give

$$
\begin{aligned}
&(A_n + \Delta A_f)(P_f - X)(A_n + \Delta A_f)^T - (P_f - X) \\
&\qquad + J(P_f - X)J^T \le \Psi < 0 \qquad (37)
\end{aligned}
$$

which indicates again from Remark 3 that $P_f - X \ge 0$ and therefore

$$
X_{ee} = [X]_{22} \le [P_f]_{22} = P_2
$$

This completes the proof of this theorem.

Remark 4. It is clear from Theorem 1 that, if the quadratic matrix inequalities (27)(28) respectively have positive definite solutions $P_1 > 0$, $P_2 > 0$, and $P_2 > 0$ satisfies

$$
[P_2]_{ii} \le \alpha_i^2, \qquad i = 1, 2, \cdots, n, \qquad (38)
$$

then the filter (9) determined by (29)-(30) will be such that: 1) the state of the augmented system (17) is mean square bounded; and 2) $[X_{ee}]_{ii} < [P_2]_{ii} \le \alpha_i^2$, $i = 1, 2, \cdots, n$. Hence, the design objective of variance-constrained robust filter with missing measurements will be accomplished. Note that the existence of a positive definite solution to (27) implies the asymptotical Schur stability of system matrix A, and the nonsingularity of A is required in the expression (29). This means, the Assumption 1 should hold.

We now briefly discuss the solvability of the quadratic matrix inequalities (27)-(28). By using the Schur Lemma (Lemma 4), we can transform (27) into the following linear matrix inequality (LMI):

$$
\begin{bmatrix} AP_1A^T - P_1 + \varepsilon MM^T + W & AP_1N^T \\ NP_1A^T & -\varepsilon I + NP_1N^T \end{bmatrix} < 0 \qquad (39)
$$

The inequality (39), together with the inequality constraint

$$
-\varepsilon I + NP_1N^T < 0, \qquad (40)
$$

are both linear on $\varepsilon > 0$ and $P_1 > 0$. Therefore, we can employ the standard LMI techniques in Gahinet *et al.* (1995) to check the solvability of the original matrix inequality (27). After P_1 is obtained, the inequality (28) becomes a standard Riccati-like matrix inequality, which is easy to solve. It is mentionable that, in the past decade, linear matrix inequalities (LMIs) have gained much attention for their computational tractability and usefulness in signal processing and control engineering Gahinet *et al.* (1995).

Remark 5. A typical feature of the present parameterization design approach is that, there exists much *explicit* freedom, such as the choices of the free parameters L ($L \in \mathbb{R}^{n\times p}$ satisfies $\Pi + LL^T < 0$), the orthogonal matrix $U \in \mathbb{R}^{p\times p}$, *etc.* This makes it possible that more performance constraints (*e.g.*, the transient requirement and reliability behavior on the filtering process) could be taken into account within the same framework.

As a summary, we give our main results as follows.

Corollary 1. If there exist a positive scalar $\varepsilon > 0$ and two positive definite matrices $P_1 > 0$, $P_2 > 0$ such that the LMIs (39)(40) and the matrix Riccati inequality (28) hold, and $P_2 > 0$ satisfies $[P_2]_{ii} \le \alpha_i^2$ ($i = 1, 2, \cdots, n.$), then the filter (9) determined by (29)-(30) will achieve the desired robust filtering performance for uncertain systems with missing measurements.

4. A NUMERICAL EXAMPLE

Consider the linear uncertain discrete-time stochastic system (1)-(2) with parameters given by

$$
\begin{aligned}
A &= \begin{bmatrix} 0.5 & 0.1 \\ 0.1 & -0.5 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \\
M &= \begin{bmatrix} 0.1 & 0.05 \\ -0.02 & 0.8 \end{bmatrix}, \quad N = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \\
W &= \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \quad V = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}.
\end{aligned}
$$

and the probability for complete observation is assumed to be 0.9.

The purpose of this example is to design the filter parameters, G and K, such that for all admissible perturbations ΔA, the augmented system (17) is mean square bounded, and the steady-state error covariance X_{ee} satisfies

$$[X_{ee}]_{11} \leq 0.8, \quad [X_{ee}]_{22} \leq 4.$$

Solving the LMIs (39)-(40) for ε, P_1, and then the Riccati-like matrix inequality (28) for P_2, we obtain

$$\varepsilon = 1.8286, \;\; P_1 = \begin{bmatrix} 5.8346 & 0.0064 \\ 0.0064 & 3.6628 \end{bmatrix},$$
$$P_2 = \begin{bmatrix} 0.7765 & 0.0052 \\ 0.0052 & 3.6983 \end{bmatrix},$$

One of the filter parameters, G, is calculated from (29) as follows:

$$G = \begin{bmatrix} 0.5437 & 0.0768 \\ 0.2040 & -1.1470 \end{bmatrix}.$$

To obtain another parameter, K, we choose $L = 0.5I_2$ such that $\Pi + LL^T < 0$ and select the orthogonal matrix U as I_2. Then, it follows from (30) that

$$K = \begin{bmatrix} 0.8040 & 0.0725 \\ 0.1246 & -0.8165 \end{bmatrix}.$$

Alternatively, to show the design flexibility, we choose U as $-I_2$, and subsequently have

$$K = \begin{bmatrix} -0.1346 & 0.0732 \\ 0.1253 & -1.3490 \end{bmatrix}.$$

5. CONCLUSIONS

In this paper, the linear filtering problem has been considered for parameter uncertain discrete-time stochastic systems where there is a nonzero probability of signal being absent in the measurement. This problem has been approached by assigning an upper bound to the steady-state error covariance, and by parameterizing the set of all filter gains that could achieve such an upper bound. It has been shown that, the problem is solvable if several linear matrix inequalities or Riccati-like matrix inequalities have positive definite solutions. In particular, the characterization of the desired filter gains has been given in terms of some 'free' parameters, and much design flexibility have been offered, which could be utilized to achieve more expected performance requirements. An numerical example has been provided to illustrate the effectiveness of the proposed design approach.

REFERENCES

Agniel, R.G. and E.I. Jury (1971) Almost sure boundedness of randomly sampled systems, *SIAM J. Contr.*, **9**, 372-384.

Chen, G (1990) A simple treatment for suboptimal Kalman filtering in case of measurement data missing, *IEEE Trans. Aerospace and Electronic Systems*, **26**, 413-415.

Chow, B.S. and W.P. Birkemeier (1990) A new recursive filter for systems with multiplicative noise, *IEEE Trans. Information Theory*, **36**, 1430-1435.

DeKoning, W.L (1984) Optimal estimation of linear discrete time systems with stochastic parameters, *Automatica*, **20**, 113-115.

Fu, M., C.E. de Souza and Z.-Q. Luo (2001) Finite-horizon robust Kalman filter design, *IEEE Trans. Signal Processing*, **49**, 2103-2112.

Gahinet, P., A. Nemirovsky, A.J. Laub and M. Chilali (1995) *LMI control toolbox: for use with Matlab*, The MATH Works Inc.

Kassel, R.J. and E. G. Jr. Baxa (1988) The effect of missing data on the steady-state performance of an α, β tracking filter, in: *Proc. Twentieth Southeastern Symposium on System Theory*, pp. 526-529.

NaNacara, W. and E. Yaz (1997) Recursive estimator for linear and nonlinear systems with uncertain observations, *Signal Processing*, **62**, 215-228.

Palhares, R.M., C.E. de Souza and P.L. Dias Peres (2001) Robust H_∞ filtering for uncertain discrete-time state-delayed systems, *IEEE Trans. Signal Processing*, **49**, 1696-1703.

Rosen, Y. and B. Porat (1989) The second-order moments of the sample covariances for time series with missing observations, *IEEE Trans. Information Theory*, **35**, 334-341.

Shaked, U., L. Xie and Y.C. Soh (2001) New approaches to robust minimum variance filter design, *IEEE Trans. Signal Processing*, **49**, 2620-2629.

Skelton, R.E. and T. Iwasaki (1993) Liapunov and covariance controllers, *Int. J. Control*, **57**, 519-536.

Stengel, R.F. (1986) *Stochastic optimal control: theory and application* (Wiley, New York)

Wang, Z. and B. Huang (2000) Robust H_2/H_∞ filtering for linear systems with error variance constraints, *IEEE Trans. Signal Processing*, **48**, 2463-2467.

Wang, Z., B. Huang and P. Huo (2001) Sampled-data filtering with error covariance assignment, *IEEE Trans. Signal Processing*, **49**, 666-670.

Wang, Z. and H. Qiao (2002) Robust filtering for bilinear uncertain stochastic discrete-time systems, *IEEE Trans. Signal Processing*, **50**, 560-567.

Yaz, E. and R.E. Skelton (1991) Continuous and discrete state estimation with error covariance assignment, In: *Proc. IEEE Conf. on Decision and Control*, Brighton, England, 3091-3092.